

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Informática

PROYECTO FIN DE CARRERA



INGENIERÍA INFORMÁTICA

**DESARROLLO DE INFRAESTRUCTURAS PARA
EL MODELADO DE USUARIOS**

Autora: Ana del Coso Santos

Octubre 2009

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

INGENIERÍA INFORMÁTICA

**DESARROLLO DE INFRAESTRUCTURAS PARA EL
MODELADO DE USUARIOS**

Autora: Ana del Coso Santos

Tutores: Fco. Javier Calle Gómez

Elena Castro Galán

Octubre 2009

A mis padres.

ÍNDICE

1. INTRODUCCIÓN Y OBJETIVOS	3
2. ESTADO DEL ARTE	7
2.1 WEB CRAWLERS.....	7
2.2 REDES SOCIALES.....	16
2.3 MODELADO DE USUARIOS	27
3. GESTIÓN DEL PROYECTO	47
3.1. WEB CRAWLER	47
3.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO	55
4. ANÁLISIS Y DISEÑO DEL SISTEMA	67
4.1. WEB CRAWLER	68
4.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO	78
5. IMPLEMENTACIÓN.....	105
5.1. WEB CRAWLER	105
5.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO	111
6. PRUEBAS	121
6.1. WEB CRAWLER	121
6.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO	122
7. CONCLUSIONES	127
8. LÍNEAS FUTURAS	131
BIBLIOGRAFÍA	133

ANEXO A. Aplicación Web Crawler.....	141
ANEXO B. Aplicación Entorno de trabajo con modelos de usuario	169
ANEXO C. Manuales de usuario	209
ANEXO D. Manuales de instalación.....	227

INTRODUCCIÓN Y OBJETIVOS DEL PROYECTO

Si no sabes dónde vas,
acabarás en otra parte

1. INTRODUCCIÓN Y OBJETIVOS

El modelado de usuario es una representación explícita de las propiedades de un usuario específico. Es usado para razonar acerca de las necesidades, preferencias o comportamiento futuro del usuario. Por lo que resulta muy útil a la hora de diseñar nuevos programas, para saber lo que el usuario va a desear del mismo y poder orientarlo hacia sus necesidades y deseos.

Los modelos que aparecen con frecuencia en los sistemas informáticos contienen algún tipo de modelo implícito de sus usuarios, pero dicho modelo no puede variarse según el uso que el usuario dé al sistema y, por tanto, tampoco una respuesta adaptada.

El modelado de usuarios permite la captura de información del usuario relevante para la explotación de la misma para poder enfocar de manera correcta el desarrollo de los productos y servicios ofrecidos. Es evolutivo acorde con los objetivos de negocio y de productos y servicios fijados e interviene en la caracterización de esos productos y servicios.

Además, resulta conveniente que la información utilizada para elaborar el modelo de usuarios sea lo más fiable posible, es decir, cuanto más real mejor, ya que, de esta forma, el modelo de usuarios obtendrá mejores resultados.

La necesidad de modelos de usuario viene dada por la personalización del marketing, es decir, ofrecer productos y servicios adaptados al cliente, según sus necesidades y deseos, el coste y la conveniencia para dicho cliente. Además, cada vez existe una mayor demanda de sistemas que se adapten a las necesidades y preferencias de los usuarios, puesto que ésta va a ser la clave que diferencie unos sistemas de otros, y que hagan al usuario decantarse por uno o por otro, debido a que la oferta de sistemas especializados es cada vez mayor.

Con todo ello se pretende mejorar la relación con el cliente, aumentar la distancia con la competencia, mejoras en la cadena de producción, satisfacción total del cliente.

Los modelos de usuarios son, cada vez más, una parte muy importante del diseño de un sistema. Puesto que la creación de un sistema viene guiada por una gestión del proyecto, que controla que el tiempo y el coste sean mínimos, surge la necesidad de crear sistemas que puedan acelerar las fases de desarrollo, haciéndolo más ‘cómodo’ posible para el desarrollador.

Para completar el capítulo se pasarán a describir los objetivos de este proyecto, que tiene como finalidad principal el análisis y posterior desarrollo de una herramienta que ayude a la definición de modelos de usuario, es decir, pretende ser un programa que facilite a sus usuarios la ardua tarea de definición de modelos de usuario. Los datos del modelo de usuarios que se va a manejar son reales, extraídos de la red social *Facebook*.

El programa deberá estar provisto de una interfaz, gráfica a poder ser, para que sea lo más intuitivo posible; ya que pretende ser una herramienta de ayuda, de complemento para la meta final de la definición de modelos de usuario, por lo que el usuario no debería ‘perder’ su tiempo en aprender a manejar un complicado programa. Además de la interfaz el sistema contará con un repositorio, donde se irá almacenando la información sobre los modelos de usuario que se van creando o modificando con el paso del tiempo de uso.

Entonces, para lograr la meta final de la definición de modelos de usuario, será necesario fijar unos sub-objetivos primarios que permitan llegar al objetivo final. Así pues se necesitará un sistema que cumpla los requisitos detallados a continuación, que una vez satisfechos, harán posible alcanzar el objetivo final:

- Adquirir información del usuario actual, permitiendo así enriquecer el posible modelo de usuario con nuevo conocimiento acerca del mismo.
- Inferir información del usuario actual, a partir de los datos contenidos en el modelo de usuario.
- Comprobar información del usuario actual, para lo cual la aplicación proporcionará una certeza que defina el grado de veracidad de la información seleccionada.
- Obtener información a partir de la inferencia del usuario actual, obteniendo el resultado de una consulta SQL.
- Cerrar la sesión, almacenando en el modelo todo el conocimiento aprendido en la misma.
- Cancelar la sesión del usuario actual, evitando así el almacenamiento del posible conocimiento aprendido en la misma.

A partir de la información proporcionada por el sistema sobre la definición de los modelos de usuario, el Usuario de dicho sistema podrá, por ejemplo, desarrollar su programa sabiendo lo que desean sus usuarios potenciales.

ESTADO DEL ARTE

Tras las huellas...

2. ESTADO DEL ARTE

En este capítulo se detalla la base teórica sobre la que se basa el presente escrito, lo cual nos permitirá determinar cómo ha sido tratado el tema de los *Web Crawlers* (o arañas de búsqueda), de las redes sociales y del modelado de usuarios hasta la fecha, cómo se encuentra ahora y cuáles son las tendencias.

Por ello se dividirá en tres grandes apartados: *Web Crawlers*, redes sociales y modelado de usuarios.

2.1 WEB CRAWLERS

2.1.1. Conocimientos previos

Algunas definiciones previas:

- URL (Uniform Resource Locator): es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización. y se refiere a la dirección única que identifica a una página Web en Internet.

- *Spambot*: tipo de araña Web que se encarga de recorrer la Web en busca de direcciones de e-mail para agregarlos a listas y luego enviarles spam.

- *Webbot*: robot de la Web.

- WWW (World Wide Web): conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto.

Un *Web Crawler* (o araña Web) es un programa que inspecciona las páginas de la World Wide Web de forma metódica y automatizada. Son un tipo especializado de *webbot* que se encarga de llevar a cabo un tipo concreto de tareas. En particular, recorre las páginas Web de Internet, las descarga al ordenador local, las parsea y las procesa.

En general, comienzan visitando una lista de urls, identifica los hiperenlaces en dichas páginas y los añade a la lista de urls a visitar de manera recurrente de acuerdo a determinado conjunto de reglas. Normalmente se le da al programa un grupo de direcciones iniciales, para

proceder a su descarga, más tarde analiza las páginas y busca enlaces a páginas nuevas. Luego descarga estas páginas nuevas, analiza sus enlaces, y así sucesivamente.

2.1.2. Orígenes y evolución

El primer Web Crawler fue muy simple, principalmente por la escala y el contenido de la Web, ya que a finales de 1993, principios de 1994 Internet era pequeño, limitado principalmente para investigación e instituciones educativas.

De hecho, el índice del primer Web Crawler contenía páginas de sólo 6.000 sitios. Tenía cuatro piezas centrales: una base de datos en la que almacenar el estado, un conjunto de agentes que recupera páginas de la Web, un operador para coordinar los agentes y la base de datos y un índice para actualizar los índices de textos completos con nuevos documentos recuperados.

Su arquitectura se puede apreciar a continuación:

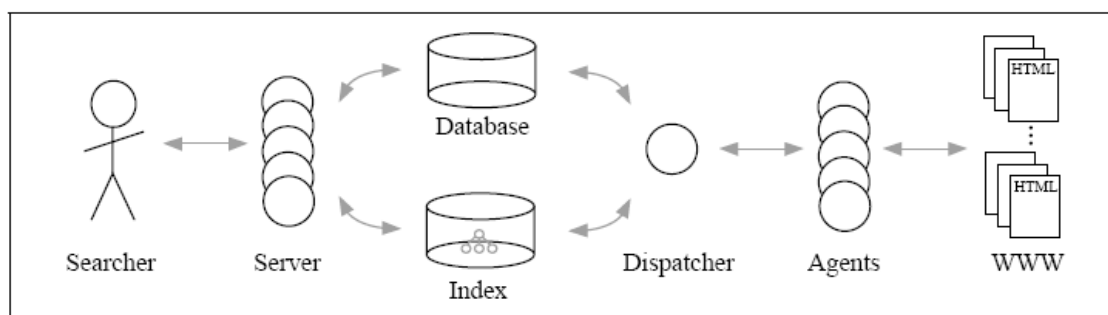


Ilustración 1. Arquitectura del primer Web Crawler

En 1994 Eichman publica el primer crawler, RBSE, cuya arquitectura se basa en dos componentes: “spider”, que actualiza la base de datos y “mite”, que descarga los documentos y los almacena como documentos ASCII.

En Enero de 1994 Brian Pinkerton, un estudiante de Informática de la universidad de Washington, comenzó un Web Crawler en su tiempo libre, World Wide Web Worm, que es el primer índice público de títulos y urls.

En 1997 Burner crea el Internet Archive, que guarda un histórico de hosts e IPs. En 1998 aparece WebSPHINX (Miller), que es el primero en utilizar multi-hilo y HTML parking. En ese mismo año, Brin y Page crean Google Crawler, donde el crawler y el sistema están integrados utilizan una arquitectura centralizada.

Es Boldi quien en 2004 crea el Ubicrawler, que es un crawler distribuido, programado en Java y con todas las funciones descentralizadas. Cada CPU reporta 660 páginas por segundo [Boldi, 2004] .

2.1.3. Funcionamiento de los Web Crawlers

Por lo general, un crawler dispone de un conjunto inicial de urls, conocidas como semillas. Entonces el crawler va descargando las páginas Web asociadas a las semillas y buscando dentro de éstas otras urls. Cada nueva URL encontrada se añade a la lista de urls que el crawler debe visitar. A este proceso se le denomina recolección de urls.

Existen distintas políticas para escoger la siguiente URL que el crawler visitará. En general, estas políticas se basan en las respuestas a preguntas tipo: ¿Cómo de importante es la página en la que estoy? ¿Cómo de importante es el sitio en el que se encuentra la página Web actual? ¿He visitado ya alguna página Web del dominio de la página a la que tengo intención de dirigirme?

Según el crawler accede a una nueva URL, la página Web asociada es descargada al ordenador local. Una vez ahí, éstas son parseadas y procesadas. Es importante mencionar que ningún crawler puede acceder a todas las urls que hay en Internet, pues el número de páginas existentes es gigantesco. Estadísticamente, el porcentaje de Internet que suele ser explorado por un crawler es aproximadamente del 15%.

Cuando el crawler parsea una página Web, lo que hace es decidir qué partes de ésta son de utilidad. Por ejemplo, puede quedarse sólo con los enlaces, sólo con imágenes, sólo con texto... según interese.

Tras el parseo, el crawler procesa la información disponible, es decir, aplica algún tipo de algoritmo para conseguir el objetivo establecido. Por ejemplo, comprobar la disponibilidad de un enlace o verificar el tamaño de una imagen.

También conviene mencionar algunas de las dificultades a las que los crawlers se deben enfrentar: enormes cantidades de páginas que recorrer, elevado número de actualizaciones de páginas existentes, páginas que crean su contenido de forma dinámica, redireccionamientos...

Algo que la mayoría de los crawler tienen en cuenta es un archivo de texto en el directorio raíz del sitio que le indica qué directorios no debe indexar. Esto último es muy importante ya que de no encontrar este archivo, los crawler indexarán todo el sitio de nuevo con toda la información que contenga.

2.1.4. Usos de los Web Crawlers

Uno de los usos más frecuentes que se les da consiste en crear una copia de todas las páginas Web visitadas para su procesamiento posterior por un motor de búsqueda que indexa las páginas proporcionando un sistema de búsquedas rápido.

Su principal uso es en motores de búsqueda, pero éste no es el único uso en absoluto. Las arañas Web (o crawlers) pueden utilizarse para multitud de tareas. Por ejemplo [Montero, 2009] :

- Invitar a todos los miembros de Tuenti que salgan de fiesta por Madrid centro y estudien en la Universidad Carlos III de Madrid
- Verificar que todos los enlaces de una página Web apuntan a páginas activas.
- Crear periódicamente un informe de cambios del sitio de una empresa competidora.
- Realizar un estudio estadístico sobre el uso de un determinado término en páginas españolas.
- Enviar un mensaje de correo electrónico cuando se añadan noticias a una página de noticias.

El uso de un crawler puede ser muy variado, estando los límites establecidos únicamente en la imaginación.

Cabe destacar que también existen crawlers con fines ilegales: spambots. Estos programas tienen un propósito malicioso y suelen recurrir a técnicas maliciosas como la falsificación de identidad para lograr sus objetivos.

La posibilidad de crear crawlers específicos, abre un nuevo mercado para los desarrolladores Web, un mercado en el que el diseño de crawlers con tareas concretas cobra gran relevancia, debido a que hoy en día, son cada vez más las empresas que desean incluir arañas Web en su negocio.

2.1.5. Políticas de búsqueda

Las webs tienen unas características que hacen difícil la tarea que desempeñan los Web crawlers, entre estas propiedades se pueden destacar tres por la mayor importancia que tienen:

- *Gran volumen de webs*, lo que implica que el Web crawler solo podrá descargar una porción de las webs en un tiempo dado por lo que es necesario una política de priorización para hacer las visitas a las distintas urls.

- *Muchos cambios de contenido*, lo que implica que cuando un Web crawler esté descargando las últimas páginas de un sitio Web es muy probable que se hayan añadido nuevas páginas, eliminado o modificado otras.

- *Generación de páginas dinámicas*, con lo que se puede producir un número de urls a visitar enorme dependiendo de los parámetros que se pasen al servidor al solicitar una página de este tipo.

El comportamiento de un Web crawler es una combinación de varias políticas, las cuales se describen a continuación:

1) POLÍTICA DE SELECCIÓN

Es un hecho que el tamaño actual de la Web es inmensamente grande, por lo que los motores de búsqueda sólo pueden abarcar una pequeña porción de Internet. Según estudios realizados sobre el tema, como mucho un buen motor de búsqueda abarcará aproximadamente un 16% de la Web, por lo que es muy interesante que los robots no descarguen páginas al azar si no que se centren en las más relevantes.

Pare ello se requiere de una métrica de la importancia para dar prioridad a las páginas webs. La importancia de una Web está en función de su calidad intrínseca y del número de visitas que tiene entre otros factores. El diseño de una buena política de selección tiene otro inconveniente que es el hecho de que el sistema completo de páginas Web a visitar no se conoce durante la ejecución del Web crawler por lo que se debe trabajar con información parcial.

2) POLÍTICA DE REVISITADO

Escanear una fracción de la Web lleva bastante tiempo y dada la naturaleza extremadamente dinámica del contenido de Internet es muy probable que en los sitios ya visitados se hayan producido modificaciones como pueden ser actualización, inserción o borrado de contenidos. Desde el punto de vista de los motores de búsqueda hay un coste asociado al no detectar un evento y es el hecho de no tener la versión más reciente del recurso que se ha modificado. Los parámetros que influyen en que un recurso esté actualizado son la frescura y la edad de la copia que se ha descargado.

● *Frescura*: Se trata de una medida binaria que indica si la copia local es exacta o no. La frescura de una página p en un instante de tiempo t se define como:

$Frescurap(t) = 1$ si p es igual que la copia local, en cualquier otro caso la función toma el valor 0.

● *Edad*: Es una medida que indica como de anticuada está la copia local de una página p en un instante de tiempo t . La edad se define como:

$Edadp(t) = 0$ si p no ha sido modificada en el instante de tiempo t , si se ha modificado la función tomará el valor $t - m$, siendo m el instante de la modificación de p .

El objetivo de un Web crawler es minimizar al máximo la edad media y mantener lo más alto posible la frescura de las páginas que ha descargado.

3) POLÍTICA DE CORTESÍA

Los Web crawlers son beneficiosos para varias tareas pero tienen una serie de inconvenientes que los hacen perjudiciales como son los siguientes:

- Los Web crawlers requieren un gran ancho de banda agotando así los recursos de la red.
- Sobrecarga del servidor, especialmente si el número de accesos al mismo es demasiado alta.

Una solución parcial a este problema es el protocolo de exclusión de robots que es un estándar para que los administradores de un sitio indiquen a los robots a qué partes del servidor pueden acceder.

4) POLÍTICA DE PARALELIZACIÓN

Un Web crawler múltiple es un robot que ejecuta múltiples procesos en paralelo. El objetivo es maximizar la tasa de descarga mientras se minimizan los efectos colaterales que produce la paralelización, así como evitar la descarga de una misma página varias veces. Para evitar este último inconveniente el Web crawler precisa de una política para asignar las nuevas urls descubiertas durante la exploración de las webs visitadas, dado que una misma URL puede ser encontrada por varios procesos del Web crawler.

Para ello existen dos tipos de políticas:

- *Asignación dinámica*: en la que existe un servidor central que se encarga de asignar las urls a visitar a los distintos crawlers, lo que permite balancear la carga que tienen los robots.

- *Asignación estática*: en la que hay una regla definida desde el comienzo de la ejecución que define como asignar las nuevas urls.

2.1.6. Web Crawlers más relevantes

1) YAHOO CRAWLER (SLURP)

Es la araña que Yahoo utiliza para indexar las páginas Web que conforman Internet. Recopila documentos para ir construyendo un índice rastreable para los servicios de búsqueda que utilizan el MDB de Yahoo!. Éstos son rastreados porque contienen enlaces que dirigen hacia ellos. Esta araña tiene en cuenta los estándares del robots.txt para asegurarse de que no se rastreen e indexan páginas que no quieren aparecer en su índice.

Debido a que Slurp busca en miles de millones de páginas en toda la Web, usan varios sistemas para rastrearlas. Por ello, se hace posible que los servidores Web puedan registrar peticiones de diferentes IPs del robot de Yahoo!.

2) GOOGLE CRAWLER

Programado en C++ y Python. Sólo se conoce la arquitectura de una versión temprana. El crawler fue integrado con el proceso de la indexación de direcciones, porque el texto que analizaba fue hecho para la indexación de direcciones con texto completo y también para la extracción de urls. Hay un servidor de urls que envía listas de urls que se recogerán por distintos crawlers. Durante el análisis, las urls encontradas son pasadas a un servidor de urls que comprueba si la URL se ha visto previamente. Si no, la URL es agregada a la cola del servidor de urls.

3) UBICRAWLER

Es un crawler distribuido escrito en Java, que no tiene ningún proceso central. Se compone de un número de “agentes idénticos”; y se calcula la función de la asignación usando el hashing consistente de los nombres de anfitrión. Hay superposición cero, lo que significa que no se recolecta ninguna página dos veces, a menos que un agente de arrastre se estrelle (entonces, otro

agente debe recolectar las páginas del agente que falla). El crawler está diseñado para alcanzar alta escalabilidad y para ser tolerante a los fallos.

4) POLYBOT

Es un crawler distribuido, escrito en C++ y Phyton. Está compuesto por un “*crawl manager*”, uno o más “*downloaders*” y uno o más servidores DNS. .”. Las urls obtenidas se agregan a una cola en disco, y se procesa más adelante para buscar urls en proceso discontinuo. La política de la cortesía considera los terceros y segundos dominios del nivel porque los terceros dominios del nivel son recibidos generalmente por el mismo servidor Web.

5) RBSE

Fue el primer Web crawler publicado. Está basado en dos programas: el primero, “*spider*” que mantiene una cola en una base de datos relacional y el segundo, “*mite*”, que es un navegador modificado de WWW ASCII que descarga las páginas del Web.

6) WEBCRAWLER

Fue utilizado para construir el primer índice con texto completo público disponible de un subconjunto del Web. Basado en la librería del CERN para acceder a varios tipos de contenido con diferentes protocolos, incluyendo HTTP, FTP y Gopher, y otro programa para analizar y para pedir urls para la exploración breadth-first del grafo del Web. También incluyó Recolector en tiempo real que sigue la ruta de navegación basándose en la semejanza del texto del vínculo con la pregunta proporcionada por el usuario.

7) WORLD WIDE WEB WORM

Uno de los primeros motores de búsqueda. Fue un crawler usado para construir un índice simple de los títulos y de urls del documento. El índice podía ser buscado usando el comando *grep* del Unix.

8) WEBFOUNTAIN

Es un crawler distribuido y modular, escrito en C++. Después de descargar las páginas, una tasa del cambio o índice de actualización se deduce para cada página y se debe utilizar un método de programación no lineal para solucionar el sistema de ecuaciones para maximizar la

frescura. Los autores recomiendan utilizar esta orden de crawling en los primeros tiempos del crawling, y después cambiar a un orden de crawling uniforme, en el cual todas las páginas se estén visitando con la misma frecuencia.

El sistema es capaz de procesar 250 millones de páginas Web a la semana, de las que extrae todo tipo de información y con la que puede tomar el pulso a cualquier tipo de información que circule sobre Internet.

El motor es capaz de estudiar la frecuencia de relación entre dos conceptos, aunque la intención es transformar a "Webfountain" en un sistema de análisis de Internet.

9) WEBRACE

Módulo de crawling implementado en Java, y usado como una parte de un sistema más genérico llamado eRACE. El sistema recibe peticiones de los usuarios para descargar páginas Web, así que los actos del crawler en parte funcionan como servidor proxy inteligente. El sistema también maneja los pedidos para suscripciones a las páginas Web que deben ser supervisados: cuando las páginas cambian, deben ser descargadas por el crawler y el suscriptor debe ser notificado.

La característica más excepcional de WebRACE es que, mientras que la mayoría de los crawlers comienzan con un sistema de la "semilla" de urls, WebRACE está recibiendo continuamente urls nuevas para comenzar.

10) COBWEB

Utiliza a un "planificador central" y una serie de "crawlers distribuidos". Los colectores analizan las páginas Web descargadas y envían las urls descubiertas al planificador, que alternadamente las asignan a los colectores. El planificador hace cumplir una orden de la búsqueda breadth-first con una política de la cortesía para evitar sobrecargar los servidores del Web. El Crawler está escrito en Perl.

11) MERCATOR

Es un crawler modular escrito en Java. Su modularidad surge del uso de los módulos intercambiables de protocolo y de los módulos de procesamiento. Los módulos de los protocolos se relacionan con cómo adquirir las páginas Web, y los módulos de procesamiento se relacionan con cómo procesar las páginas Web. El módulo de proceso estándar sólo analiza las

páginas y extrae nuevas urls, pero otros módulos de proceso se pueden utilizar para poner en un índice el texto de las páginas, o para recopilar estadística de la Web.

2.2 REDES SOCIALES

2.2.1. Conocimientos previos

Una red social es una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos (a veces denominados actores) y las aristas relaciones entre ellos.

En las redes sociales en Internet se presenta la posibilidad de interactuar con otras personas aunque no las conozcamos, el sistema es abierto y se va construyendo obviamente con lo que cada suscrito a la red aporta, cada nuevo miembro que ingresa transforma al grupo en otro nuevo. La red no es lo mismo si uno de sus miembros deja de ser parte.

Intervenir en una red social empieza por hallar allí otros con quienes compartir intereses, preocupaciones o necesidades y aunque no sucediera más que eso, eso mismo ya es mucho porque rompe el aislamiento que suele aquejar a la gran mayoría de las personas, lo cual suele manifestarse en retraimiento y otras veces en excesiva vida social sin afectos comprometidos.

Las redes sociales en Internet suelen posibilitar que varias personas se conjuguen en comunidades y quizás allí esté gran parte de toda la energía que le da vida a los grupos humanos que conforman esas redes. Las redes sociales dan al anónimo popularidad, al discriminado integración, al diferente igualdad, al malhumorado educación y así muchas cosas más.

La fuerza del grupo permite sobre el individuo cambios que de otra manera podrían ser difíciles y genera nuevos vínculos afectivos y de negocios.

Básicamente el funcionamiento comienza cuando una vez montado el soporte técnico, un grupo de iniciadores invitan a amigos y conocidos a formar parte de la red social, cada miembro nuevo puede traer consigo muchos nuevos miembros y el crecimiento de esa red social puede ser geométrico. Y he aquí que se transforma en un interesante negocio.

2.2.2. Orígenes y evolución

El primer sitio de redes sociales reconocibles puesto en marcha en 1997 fue *SixDegrees.com* y permitía a los usuarios crear perfiles, lista de amigos y amigos de sus amigos. De 1997 a 2001, *AsianAvenue*, *Blackplanet* y *MiGente* permitían a los usuarios crear relaciones personales y profesionales, creando perfiles que permitían a los usuarios identificar amigos en sus redes sin pedir la aprobación de esas conexiones.

Desde entonces se han creado diversas redes: unas permanecen y otras han desaparecido, según la zona geográfica el líder puede ir cambiando, pero en el presente año (2009) los principales competidores a nivel mundial son: *MySpace*, *Facebook*, *LinkedIn*, *Twitter*.

Ryze.com se inició en 2001 para ayudar a las personas aprovechar sus objetivos empresariales y profesionales mediante la creación de redes profesionales. Su premisa fue proporcionar una red de apoyo para los consumidores en un entorno profesional que les permita la búsqueda de puestos de trabajo, hacer las conexiones de empresas, y potenciar las comunidades virtuales para las necesidades de las empresas en general. *LinkedIn* también fue creado para capturar este segmento de mercado y ha evolucionado hasta ser el estándar de la industria profesional para la creación de redes con muchas de las características de redes sociales adoptadas en el sitio.

MySpace se instaló en 2003 y se ha diferenciado de otros sitios por que permite a los usuarios personalizar sus páginas. Los adolescentes fueron los primeros en adoptarlo y *MySpace* floreció a través de la atracción de este grupo demográfico. *MySpace* incluso ha modificado su política de usuario para permitir a los menores de edad crear sus propios perfiles a raíz de la demanda en el segmento de mercado. Conforme el sitio ha crecido, tres poblaciones se empezaron a formar: los músicos / artistas, jóvenes, y adultos en su etapa posterior a la universidad.

Google lanzó en enero de 2004 *Orkut* apoyando un experimento que uno de sus empleados realizaba en su tiempo libre. En 2005 ingresan *Yahoo 360°* y otros.

Facebook fue creado originalmente para apoyar a las redes universitarias, en 2004 los usuarios del sitio estaban obligados a proporcionar las direcciones de correo electrónico asociada con las instituciones educativas. Este requisito lo llevo a ser un modelo de acceso relativamente cerrado y por lo que los usuarios tenían la percepción de la Web como un sitio íntimo y privado de una comunidad demográfica cerrada. Posteriormente se amplió para incluir a los estudiantes de secundaria, profesionales, y finalmente todos los usuarios potenciales de

Internet. A diferencia de otros, en *Facebook* los usuarios sólo pueden hacer públicos sus perfiles a otros usuarios del sitio. Otra característica que distingue a Facebook es la capacidad para desarrolladores externos de crear aplicaciones lo que permite a los usuarios personalizar sus perfiles y realizar otras tareas, tales como comparar las preferencias de las películas [Boyd-Ellison, 2007] .

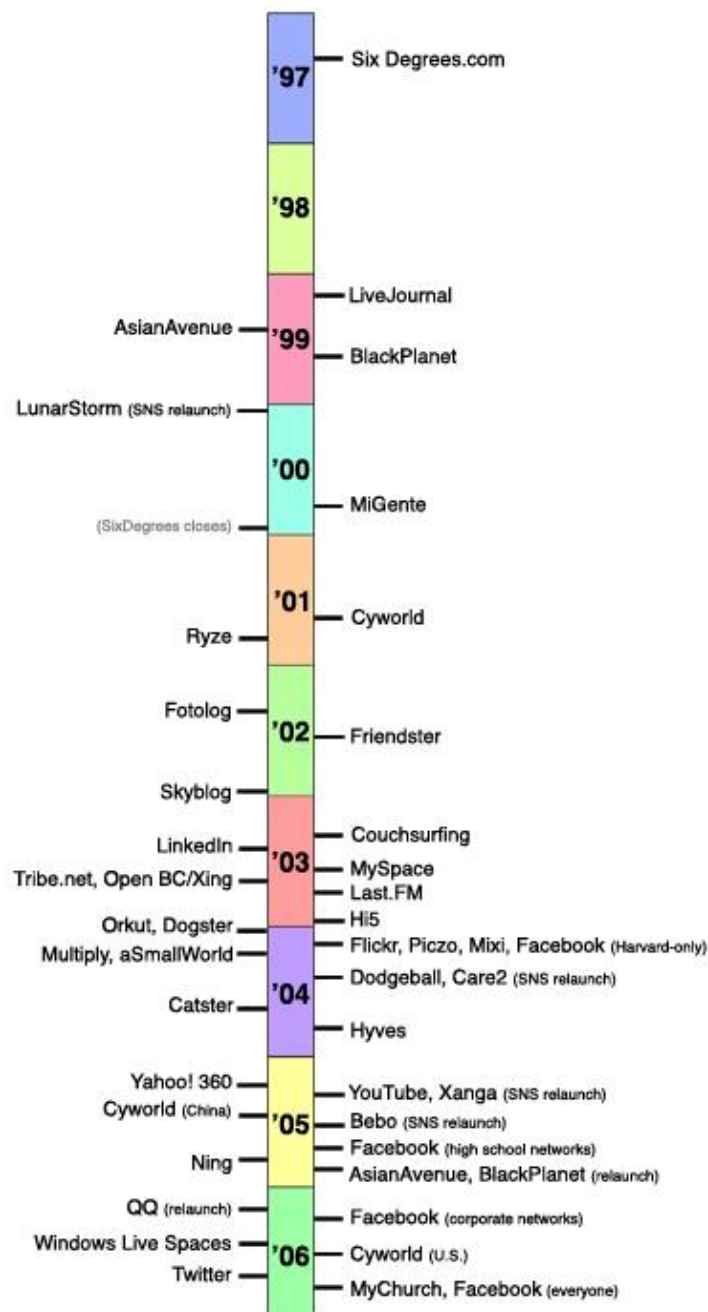


Ilustración 2. Cronología de las redes sociales

2.2.3. Tecnología de las redes sociales

Una revisión de los sitios de redes sociales hace evidente que sus fundamentos tecnológicos están basados en la consolidación de aplicaciones de uso común en un único sitio. Se emplean las tecnologías estándares, como el correo electrónico y sus protocolos; http para facilitar las operaciones de subir o bajar información ya sea fotos o información sobre el perfil; la mensajería instantánea y los otros protocolos de uso común que se pueden encontrar a lo largo de muchas páginas Web.

Las características de las salas de Chat también están disponibles, y permiten a los usuarios conectarse instantáneamente en modalidad de one-to-one o en pequeños grupos en función de la oferta del sitio.

Los principales proveedores de servicios de redes sociales están contruidos sobre plataformas que apoyan las necesidades de los consumidores con diferentes herramientas para facilitar el intercambio de información. Estas tecnologías se encuentran en una etapa de madurez, se usan de forma generalizada por lo que son consideradas un componente integral de muchos de los principales sitios. El reciente despliegue de la nube de computación y las ofertas de almacenamiento han permitido a los proveedores ofrecer los servicios para satisfacer su creciente demanda.

Los dispositivos móviles se están convirtiendo en una fuente primaria de acceso significativo los consumidores de estas redes sociales. Esto ha sido fundamental en el cumplimiento de los requisitos de conectividad que impulsa muchas de las características de redes sociales. Como se trata de mercados maduros probablemente habrá innovaciones que seguirán impulsando las nuevas características y capacidades que se adapten a la satisfacción de necesidades de este mercado demográfico. Para los servicios de redes sociales no existen normas específicas para el despliegue de la tecnología de cada sitio. Sin embargo, como todos ellos son ofrecidos a través de Internet utilizando navegadores Web, deben atenerse a las normas comunes que son impulsadas por la aplicación y el diseño funcional de las necesidades. Por lo tanto, las normas son inducidas más por el mercado y menos por un consejo de administración o por reglamentos oficiales.

A medida que los nuevos navegadores y plataformas se han desplegado los proveedores están obligados a adaptar e integrar y a permitir que sus sitios sigan siendo utilizables por nuevos consumidores.

2.2.4. Redes sociales más relevantes

1) Badoo

Creado en la ciudad de Londres en 2006 de la mano de un grupo de jóvenes programadores y emprendedores tecnológicos. La idea era crear una red social de Internet que pudiera romper barreras a nivel mundial, pero que al mismo tiempo, resguardara de manera local la identidad cultural de los usuarios. El objetivo era permitir a la gente compartir sus vidas y atraer la atención de otras personas a nivel global y local. Hoy Badoo se ha convertido en una de las principales 300 webs del mundo según Alexa y goza de una gran popularidad como red social en los mercados europeos y americanos. Badoo continua expandiéndose rápido y cuenta con una base de 14 millones de usuarios en el mundo, procedentes de 180 países diferentes.

La red social Badoo está disponible en 16 idiomas y cuenta actualmente con más de 40.000 nuevos usuarios por día, y entre 50.000 y 200.000 usuarios conectados a cualquier hora.

2) Bebo

Bebo es el acrónimo de "Blog early, blog often" es una red social fundada en enero de 2005. Bebo es similar a otros sitios de redes sociales, te permite compartir fotos, enlaces, vídeos, aficiones e historias con quien quieras desde un solo lugar, conectar con amigos, familiares, compañeros de clase o de trabajo y nuevas amistades (aunque sean miembros de otra red social), recibir recomendaciones sobre música, vídeos, artículos y juegos, conocer a otros usuarios con gustos similares, dibujar en una pizarra virtual o en la de otros miembros. Fue comprado por AOL el 13 de marzo de 2008 por 850 millones de dólares.

Bebo es una red social y es posible acceder a través de Internet y a través de servicios WAP ofrecidos por ciertos proveedores de telefonía móvil.

3) BlackPlanet

Es un sitio de red social que tiene por objetivo principal la comunidad americana y africana. Fue lanzado el 1 de Septiembre de 1999 por Omar Wasow.

En Diciembre de 2007 se situó en la cuarta posición de sitios de redes sociales según el sitio de monitorización de Internet Hitwise. En Junio de 2007 tenía 16.500.000 de miembros y 80.000.000 de páginas.

En Abril de 2008, Radio One, adquirió BlackPlanet y su relación con la compañía Community Connect Inc. por 38 millones de dólares.

4) CLASSMATES.COM

Es un servicio de red social creado en 1995 por Randy Conrads, quien fundó Classmates Online Inc. El sitio Web ayuda a encontrar miembros, a conectarse y a mantener en contacto a amigos y gente que hayan ido conociendo en su vida (incluyendo compañeros de guardería, escuela primaria, instituto, universidad, trabajo y servicio militar). Tiene más de 40 millones de miembros activos en Estados Unidos y Canadá.

5) FACEBOOK

Facebook es una herramienta social que pone en contacto a personas con sus amigos y otras personas que trabajan, estudian y viven en su entorno. Es accesible por todos los usuarios de Internet desde septiembre de 2006.

Originalmente fue creado para estudiantes de la Universidad de Harvard. Ofrece servicios tales como buscador de amigos, creación de grupos (para reunir personas con intereses comunes), escribir en el muro de amigos (y que ellos te escriban) y un sin fin de aplicaciones diseñadas exclusivamente para este sistema.

Cabe destacar la funcionalidad de que el usuario pueda escribir un mensaje de estado y que todos sus amigos lo vean y comenten sobre él.

6) FOTOLOG

Fue creado en Mayo de 2002 como Fotolog.net y es el sitio Web publicación de fotografías más grande del mundo, superando incluso a Flickr. Está dedicado a blogs fotográficos conocidos más genéricamente como fotologs.

En diciembre de 2005 el sitio cambió su nombre a Fotolog.com. En Septiembre de 2008 Chile es el país con más cuentas creadas en Fotolog.com, con un total aproximado de 4,8 millones.

Actualmente tiene más de 20 millones de usuarios. Sus creadores son Scott Heiferman, Adam Seifer y una tercera persona conocida sólo como Spike.

7) HABBO

Habbo (o Habbo Hotel) es una de las más grandes redes sociales en Internet, operada por Sulake Corporation. Está enfocado a jóvenes y adolescentes y presenta salas de Chat con formas de habitaciones de un hotel. El registro y la entrada a Habbo es totalmente gratuito, pero el acceso a servicios adicionales requieren la compra de "Habbo Créditos" con dinero real, que es la moneda dentro del Hotel.

El servicio fue lanzado en el año 2000, se ha llegado a expandir a 31 países y está disponible en 20 idiomas. El crecimiento de la red es tan grande que diariamente se unen más de 75.000 usuarios nuevos, se registran 9,5 millones de visitantes únicos al mes y existen más de 100 millones de cuentas creadas. Además, el 90% de los visitantes tiene entre 13 y 18 años.

En 2008, Habbo contaba con más de 104 millones de cuentas creadas a nivel mundial, donde el hotel español y latinoamericano proporciona más de 15 millones de cuentas a la cifra total. La cantidad de usuarios conectados simultáneamente más alta de todos los hoteles se ha registrado en el de España y Latinoamérica, donde han llegado a jugar aproximadamente 58.000 usuarios al mismo tiempo.

8) HI5

Se trata de una red social basada en un sitio Web lanzada en 1996. En el momento de crear o modificar una cuenta de hi5, el usuario puede configurar su propia cuenta, llenando los espacios de su información con la del usuario propio para que así toda la red de amigos pueda informarse.

Todos los usuarios que acceden a una cuenta podrán ver toda la información que se haya registrado. Cabe expresar que el usuario dueño del perfil, puede restringir el acceso al mismo, dejando libre entrada a amistades en su red de amigos.

Como funcionalidades se pueden destacar los amigos que puede tener una cuenta, realizar comentarios, conocer las visitas que han pasado por tu página/perfil, creación de un diario, además de otras aplicaciones externas creadas para este sistema.

9) MYSPACE

Es un sitio Web de interacción social formado por perfiles personales de usuarios que incluye redes de amigos, grupos, blogs, fotos, vídeos y música, además de una red interna de mensajería que permite comunicarse a unos usuarios con otros y un buscador interno.

MySpace es el séptimo sitio más visitado de Internet y el cuarto sitio más visitado de la red de lengua inglesa; aunque por otro lado, este sitio es poco frecuentado en Europa. Se lanzó en 2003.

Como características se pueden describir los boletines, los grupos, servicio de mensajería instantánea, compartimiento de videos y otra serie de aplicaciones diseñadas para MySpace.

Dentro del perfil de un usuario se pueden incluir las siguientes secciones: emoticonos que demuestran el humor, secciones (acerca de mí, intereses, etc.), comentarios, personalización y música.

10) NETLOG

Es una Web de red social belga diseñada especialmente para los jóvenes europeos. Fue fundada en 2004.

Los miembros de esta comunidad pueden crear su propia página Web, extender su red social, publicar sus propias listas de música, compartir videos, postear blogs e incluirse en grupos llamados 'clans'.

Cada miembro puede invitar a otros desde su página, momento en el cual se crea un link entre ambos perfiles.

11) ORKUT

Orkut es una red social promovida por Google desde enero del 2004. La red está diseñada para permitir a sus integrantes mantener sus relaciones existentes y hacer nuevos amigos, contactos comerciales o relaciones más íntimas.

Es posible crear y mantener comunidades, que agrupan personas de acuerdo a sus gustos e intereses, en diferentes categorías, entre las que se cuentan: actividades, negocios, juegos, música, mascotas, religión, escuelas, comidas, preferencias sexuales, y algunas más.

Hasta hace algún tiempo solo podían acceder a esta comunidad aquellos que recibían una invitación de alguien que ya pertenecía a ella, esto mientras el servicio permanecía en fase beta. Hoy está abierta a cualquier persona, para ello es necesario tener una cuenta de e-mail simplemente.

12) SONICO

Sonico es una red social online de acceso gratuito orientada al público latinoamericano. Los usuarios pueden buscar y agregar amigos, actualizar su perfil personal, administrar su privacidad, subir fotos y videos de YouTube, organizar eventos, desafiar a otros usuarios en 6 juegos multijugador e interactuar con otra gente a través de mensajes privados, comentarios, marcas en las fotos y un mensajero instantáneo.

Sonico fue lanzado en julio de 2007 resaltando la importancia de tener usuarios legítimos antes que una gran cantidad de cuentas anónimas. En consecuencia, su lema es “gente real, conexiones reales”. A diferencia de otras redes sociales, Sonico tiene una llamada moderación proactiva que consiste en un equipo dedicado exclusivamente a chequear cada perfil nuevo y el contenido subido al sitio. Afirman poder gestionar más de 200.000 perfiles diarios.

Un año después de su fundación, Sonico tiene un equipo de 88 personas a la vez que sus responsables afirman estar recibiendo alrededor de 110.000 nuevos usuarios por día y reportan más de 20 millones de usuarios registrados. En el ranking de Alexa está en el puesto 329 (en Julio de 2009).

13) TUENTI

Tuenti es una herramienta que facilita la transmisión de información entre gente que se conoce. Fue lanzada en febrero de 2006. La meta es que los usuarios puedan estar al corriente de todo lo que está pasando en su círculo de amigos.

Permite al usuario crear su propio perfil, subir fotos y videos y contactar con amigos, además de poder comentar todo lo que sus contactos publiquen. Se caracteriza también por su potente buscador, que puede localizar a todos los miembros de la red por su nombre real, pudiendo introducir otros filtros para restringir los resultados de búsqueda.

Es llamado por muchos “el Facebook español”, a pesar de que difieren en muchas funcionalidades.

14) WINDOWS LIVE SPACES

Windows Live Spaces, es uno de los servicios que ofrece Windows Live como plataforma para blogs. El sitio fue publicado a principios del 2004 bajo el nombre de MSN Spaces para competir con otros sitios. Desde agosto de 2007 Windows Live Spaces recibe 27 millones de visitantes al mes, ubicándolo en el segundo puesto en el ranking de redes sociales.

Windows Live Spaces, es un servicio con el objetivo de permitir a sus usuarios expresarse libremente, ya sea con la posibilidad de publicar sus pensamientos, sus fotografías o sus intereses, con esa meta Windows Live Spaces es una opción más dentro de la blogosfera.

Dentro de sus características se encuentra la posibilidad de agregar pequeñas aplicaciones, llamadas gadgets, que permiten que el sitio sea más atractivo al usuario. Además, ofrece la posibilidad de incorporar listas de películas, de música o personalizadas; la creación de un perfil.

Entre sus funcionalidades cabe destacar: blogs, imágenes, listados (incluyendo listas de música, listados de películas, de libros de listas) y amigos (incluyendo etiquetas y notas sobre amigos), perfil (con información, datos personales, trabajo, etc.).

A continuación se va a presentar una gráfica comparativa con todas las redes sociales comentadas anteriormente. Están ordenadas según el ranking Alexa, cuyo enlace se encuentra disponible en el apartado de Referencias, y se puede apreciar el número de usuarios de cada una de ellas.

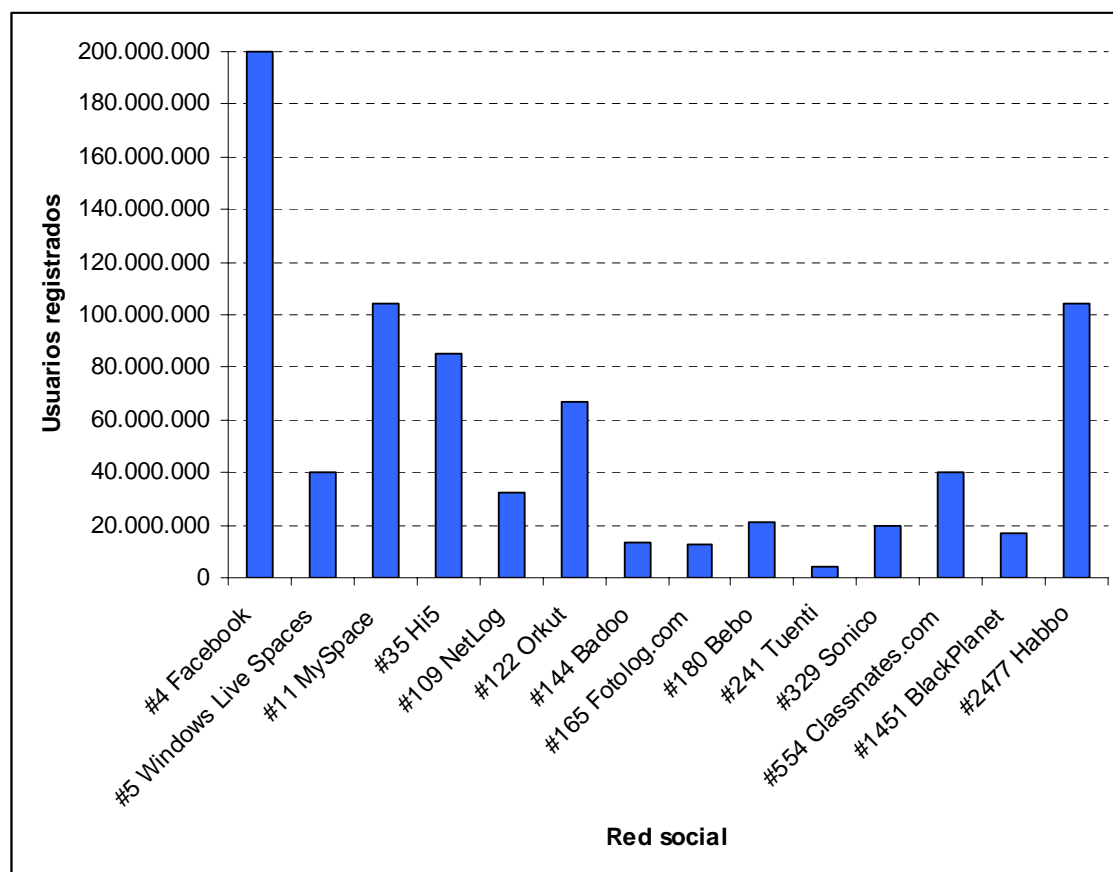


Ilustración 3. Comparación redes sociales

Para finalizar se van a comentar las razones que han llevado a elegir Facebook como red social para la obtención de la información necesaria para la aplicación a desarrollar. Primeramente se ha tenido en cuenta el número de usuarios que, tal y como se puede observar en la gráfica anterior, es sensiblemente mayor que cualquier de sus competidores. Además, permite el acceso a cierta información acerca de sus usuarios sin estar registrado previamente; dicha información contiene características propias de cada usuario, que es exactamente lo que se buscaba.

2.3 MODELADO DE USUARIOS

2.3.1. Conocimientos previos

Se entiende por modelado de usuarios una técnica utilizada para conocer los intereses, hábitos, comportamiento y necesidades de los usuarios en un sistema, situación y tiempo determinados. En otras palabras, es una representación explícita de las propiedades de un usuario específico. La característica explícito en esta definición es importante, puesto que casi todos los sistemas informáticos contienen algún tipo de modelo implícito de sus usuarios (se sabe lo que tienen que hacer los usuarios con el programa y se actúa o se programa en consecuencia). No obstante, esos modelos no son explícitos, no pueden variarse según el uso que el usuario dé al sistema y, por tanto, tampoco una respuesta adaptada.

La función del modelado de usuarios es garantizar que las habilidades, capacidades y necesidades humanas sean tomadas en cuenta en el diseño de cada componente de la interfaz.

El modelo de usuario contiene toda la información que el sistema conoce sobre el usuario. Es mantenido por el sistema, a pesar de que el usuario está capacitado para revisar y editar su perfil. Las acciones del usuario y los eventos en varios niveles conceptuales, como por ejemplo los clics del ratón, son enviados por la interfaz de usuario al perfil de usuario.

Para abordar el tema se considerarán previamente otros aspectos como:

1) INTERACCIÓN HOMBRE-COMPUTADORA

La interacción hombre-computadora comprende todo lo que ocurre cuando un hombre y un sistema computarizado realizan tareas juntos; esto involucra tanto al rol de programador como al rol de usuario final, en un diálogo hombre-computadora en el que media una interfaz.

Debido a que el enfoque de la interacción hombre-computadora era centrado en la construcción de sistemas más usables, a menudo se reducía el poder expresivo de los sistemas para acomodarlos a usuarios novatos o casuales, asumiendo que éstos usaban el sistema por primera vez. Los sistemas complejos para uso profesional deben ser útiles, deben permitir a sus usuarios hacer las tareas para obtener trabajos correctamente terminados. Estos sistemas son a menudo complicados de usar al principio, pero luego permiten a sus usuarios moverse en un gran rango de tareas. Suposiciones genéricas sobre usuarios puede ser adecuadas en sistemas para principiantes, pero sólo si hacemos caso de las exigencias para proporcionar el acceso

universal para la gente con capacidades diferentes. Suposiciones genéricas sobre usuarios experimentados está limitando la capacidad de aprendizaje y uso de estos sistemas.

Por todo ello se puede concluir que para que la comunicación entre hombre y computadora sea efectiva debe existir un conocimiento mutuo. Por un lado el usuario debe conocer el funcionamiento de la interfaz a un nivel operativo, implicando además una situación de aprendizaje y la implicación de procesos cognitivos. Por otra parte la interfaz debe estar diseñada en función de las características de ese usuario o adecuarse a distintos tipos de usuarios. Uno de los caminos para lograr esto es el Modelado de Usuarios.

Uno de los nuevos objetivos de la interacción hombre-computadora concierne a las tareas, explicaciones, justificaciones sobre las acciones, no sólo sobre las interfaces. El nuevo reto es mejorar la forma en que la gente usa los ordenadores para casi todo.

2) RECUPERACIÓN DE INFORMACIÓN

Es la ciencia de la búsqueda de información en documentos, búsqueda de los mismos documentos, la búsqueda de meta datos que describan documentos, o, también, la búsqueda en bases de datos, ya sea a través de Internet, Intranet, para textos, imágenes, sonido o datos de otras características, de manera pertinente y relevante [Robertson, 1997] .

Es un claro ejemplo donde la interfaz juega un papel de intermediario entre el usuario y el objetivo que persigue. Los ítems abordados por esta área comprenden: cómo representar y organizar la información intelectualmente, cómo especificar una búsqueda intelectualmente y cuáles son los sistemas y técnicas utilizadas para estos procesos.

En este contexto una interfaz debe posibilitar representar las necesidades de información del usuario, buscar la información apropiada para esas necesidades particulares y evaluar la efectividad de los resultados obtenidos. Para ello debe tener en cuenta:

- El análisis del problema o consulta planteado por el usuario.
- La elección de una estrategia de respuesta según el tipo de problema o consulta.
- La función de posibilitar el diálogo entre sistema y usuario.
- La función de inferir la evaluación del usuario sobre la correspondencia de los resultados logrados con el problema, la necesidad de modificar la pregunta inicial o finalización si el usuario está satisfecho.

- La realización de la vuelta sobre el análisis de la pregunta y selección de nuevas respuestas si fuera necesario.

3) MODELO COGNITIVISTA DE LA INFORMACIÓN

Es desarrollado a partir del modelo propuesto por Nicholas J. Belkin [Belkin-Oddy, 1982]. Este autor sostiene que toda interacción ya sea hombre-hombre u hombre-computadora es mediada por el estado de conocimiento que cada una de las partes tiene sobre sí misma y sobre la otra.

Sus estudios se centran en los procesos comunicativos por los que la información es transmitida y los procesos cognitivos implicados en el proceso de recuperación de la misma. Sostiene que una necesidad de información surge de una anomalía reconocida en el estado de conocimiento del usuario y que, en general, éste es incapaz de especificar en forma precisa lo que necesita para resolverla. La fuerza conductora de una situación de recuperación de información, o en nuestro caso, del uso de una interfaz para lograr un objetivo determinado, es una “situación problema”. El uso de una interfaz puede responder a la resolución de una “situación problema”, ya sea el extraer dinero de un cajero automático, escribir una carta, realizar una búsqueda en una base de datos, etc.

Solucionar esta “situación problema” implica, por un lado, tener información sobre qué se desea, y por otro, bajo qué condiciones, por medio de qué herramientas y con acceso a qué recursos se obtendrá. Representando esta información tendremos definido el problema. No obstante, las necesidades de información del usuario no son siempre especificables en forma precisa.

Las interfaces más difundidas (por ejemplo, la mayoría de las disponibles en la Web), recurren al principio del “best-match” donde, dada la representación de una consulta de información (pregunta o grupo de términos) la respuesta del sistema será la información (imagen, sonido texto, datos) que mejor se adapte a ella. Para que esto sea efectivo el usuario debe ser capaz de especificar todos los aspectos relevantes de su problema o necesidad de información en una forma equivalente a como la información está representada en el sistema.

Frente a esto Belkin propone un modelo alternativo: partiendo de entrevistas donde el usuario describe una “situación problema” se llega, mediante un análisis de contenido, a una representación estadística y descripción de sus necesidades a través de “imágenes estructurales” de áreas de interés.

El programa opera sobre una gráfica cuyos puntos representan documentos, materias y autores, las líneas representan asociaciones. Una imagen estructural del área de interés para el usuario es mantenida durante el diálogo en línea. Esta imagen no es una representación de las necesidades del usuario, sino un contexto formal dentro del cual se localizaría aquella información relevante a las mismas.

La importancia de esta propuesta está en considerar para el diseño de una interfaz a aquellos usuarios con necesidades no especificables.

2.3.2. Orígenes y evolución

En numerosas ocasiones se considera el trabajo de Elaine Rich en el sistema GRUNDY [Rich, 1979] como uno de los primeros que marcó el comienzo de esta área de investigación, y donde se comenzó a ver la posibilidad de que los ordenadores empezarán a considerar a cada usuario de manera diferente, teniendo en cuenta sus características, preferencias y objetivos. En el sistema GRUNDY a cada usuario se le asignaba un *estereotipo*, término que se utiliza para denominar a los modelos de grupos de usuarios que compartían ciertas características o intereses comunes. A cada estereotipo se le asigna una serie de propiedades o inferencias (como por ejemplo, una serie de reglas), que se aplicarán a cada usuario perteneciente a dicho estereotipo. La asignación de un usuario a un determinado estereotipo se hace en función de las características de dicho usuario (normalmente obtenidas a partir de formularios de entrada) y de la aplicación de una determinada regla de asignación (por ejemplo, si un usuario está en primer curso de carrera entonces le asignamos el estereotipo de *principiante* en dicha carrera). Esta alternativa es adecuada sólo para ciertos dominios (entornos cerrados) ya que, tal y como veremos más adelante, esta representación presenta ciertos inconvenientes a la hora de aplicarlo a dominios abiertos que engloban a grupos heterogéneos de alumnos, en los que, aunque se definan ciertos estereotipos, la asignación a los mismos de nuevos usuarios debe hacerse dinámicamente.

Desde la primera definición de modelos de usuario basados en estereotipos, el campo del modelado del usuario ha progresado notablemente en los últimos años y las áreas de aplicación donde se utilizan estos modelos es cada vez mayor. Los modelos de usuario pueden tomar muy diversas formas dependiendo de los objetivos para los están pensados. Así, podemos encontrar modelos cuyo objetivo es describir:

- Las características generales de los usuarios (casi todos los sistemas guardan esto).
- Los objetivos del usuario, como por ejemplo el sistema PUSH [Monden, 1983] .
- Los patrones de comportamiento y preferencias de los usuarios. Esta información es la base de cualquier sistema de recomendación, como por ejemplo en comercio electrónico.
- Las destrezas de los usuarios, como por ejemplo el sistema AVANTI.
- etc.

2.3.3. Tipos de modelos de usuarios

Podemos distinguir una gran variedad de tipos de modelos de usuario, los cuales pueden clasificarse teniendo en cuenta:

- El tipo de usuarios que se modelan: usuarios estándares (estereotipos) o modelos más individualizados.
- Fuentes de información para realizar el modelado: modelos construidos explícitamente por el usuario (estos modelos se construyen a partir de los datos que proporciona el usuario mediante formularios de entrada) o abstraídos por el sistema basándose en el comportamiento del usuario (datos de interacción fundamentalmente).
- La caducidad de la información que se guarda en el modelo: modelos a corto plazo con información muy específica y modelos a largo plazo con información más general.
- Actualización del modelo: modelos estáticos y modelos dinámicos.

Las técnicas de actualización de los modelos generalmente dependen del tipo de modelo del que se trate. Así por ejemplo, los modelos individualizados, los que se basan en el comportamiento de los usuarios o los modelos a corto plazo, por lo general requieren actualizaciones dinámicas. Si el modelo contiene información a muy corto plazo entonces estamos hablando de un modelo de tareas, puesto que sólo se refiere a la tarea que está realizando el usuario en ese momento, y el resto de características del usuario (objetivos generales, conocimiento de partida, etc.).

El tipo más básico de modelo que podemos considerar es el modelo estático con un usuario estándar (canonical). Este tipo de modelo se puede incorporar a un sistema fácilmente (de hecho son representaciones explícitas de los modelos implícitos mencionados anteriormente). Por el

contrario, si lo que queremos es modelar cada uno de los usuarios individualmente entonces se necesitan actualizaciones dinámicas y son necesarios métodos explícitos que describan el estado del modelo de usuario.

2.3.4. Usos del modelado de usuarios

En este apartado se mostrarán distintos ejemplos de para qué resulta útil el modelado de usuarios. El modelado de usuarios resulta necesario para convertir un marketing de masas en un marketing relacional, es decir, personalizado; para mejorar la relación, para aumentar la barrera del cambio a la competencia, para mejorar nuestra cadena de producción, para satisfacer al cliente...

En el contexto de este proyecto, los modelos de usuarios son definidos como modelos que sistemas tienen usuarios que residen dentro de un ambiente computacional. Ellos podrán ser diferenciados por modelos mentales que los usuarios tienen en los sistemas y tareas que residen en la cabeza de los usuarios, en interacciones con otros y con artefactos.

Los modelos de usuarios pueden dividirse en tres categorías:

1) REPRESENTACIÓN JERÁRQUICA DE LAS TAREAS DEL USUARIO Y DEL OBJETIVO.

Las técnicas GOMS (*Goal, Operators, Methods, Selection rules*) [Reitman-M. Olson, 1995] resultan realmente útiles para los casos en que conocemos la secuencia de operaciones y queremos encontrar rápidamente la manera en que la secuencia puede ser representada por un usuario experimentado. Ahora nos detendremos a comentar cada componente de la técnica:

- **GOALS:** los objetivos que el usuario quiere conseguir. Frecuentemente son subdivididos en sub-objetivos y analizados por separado. Por ejemplo, como tener objetivo editar un documento y como sub-objetivos mover-texto, borrar-frase o insertar-palabra.
- **OPERATORS:** un operador es una acción desarrollada para dar servicio a un objetivo.
- **METHODS:** son una secuencia de operadores y requisitos del sub-objetivo para conseguir el objetivo. Por ejemplo si el objetivo es borrar-palabra, éste podrá ser alcanzado mediante los operadores mover-ratón, hacer clic-ratón y pulsar-tecla-del, por tanto la secuencia es un método para conseguir el objetivo.

- **SELECTION RULES:** Si hay más de un método disponible para un cierto objetivo, existe la necesidad de seleccionar reglas para representar el conocimiento del usuario.

Con GOMS podemos movernos por cualquier secuencia, deduciendo el tiempo necesario aproximado para el siguiente escalón, y estimar el tiempo total a partir de lo anterior.

Esta técnica no resulta siempre apropiada, pues no ofrece respuestas exactas cuando el método de operación es desconocido o se trata de un usuario no experimentado.

2) MODELOS DE MECANISMOS FÍSICOS.

Keystroke Level Model (KLM), este modelo del nivel de pulsación de teclas usa el conocimiento relativamente fuerte en sistemas con motor humano como una base para predicciones detalladas sobre el funcionamiento del usuario.

No es capaz de modelar la fase de adquisición de la tarea del usuario, donde el usuario construye una representación mental de la tarea. La fase de ejecución del modelo se descompone en cinco operadores físicos distintos, un operador mental y un operador de respuesta del sistema.

Este modelo debe usarse con cautela, ya que las aproximaciones dadas por dicho modelo pueden producir resultados imprecisos.

3) MODELOS LINGÜÍSTICOS Y GRAMATICALES.

La interacción del usuario con un ordenador a veces se observa en términos de lenguaje. La omnipresente BNF (*Backus Normal Form*, Forma normal de Backus) [Chomsky, 1957] ha sido adoptada en los sistemas de descripción de diálogos. Por ejemplo, imaginemos un sistema de gráficos que tiene una función de dibujar línea. El primer usuario tiene que seleccionar la 'línea' de la opción del menú. La función dibujar línea permite al usuario dibujar una línea polimórfica, que tiene una secuencia de arcos con puntos. El usuario selecciona los puntos haciendo clic con el ratón en el área de dibujo. El usuario emplea doble clic para indicar que se trata del último punto de la línea. Y la forma normal de Backus para la tarea sería:

```
darw-line      ::=  select-line + choose-points
                  + last-point
select-line    ::=  position-mouse + CLICK-MOUSE
choose-points  ::=  choose-one
                  |  choose-one + choose-points
choose-one     ::=  position-mouse + CLICK-MOUSE
last-point     ::=  position-mouse + DOUBLE-CLICK-MOUSE
position-mouse ::=  empty | MOVE-MOUSE + position-mouse
```

Ilustración 4. Gramática en BNF para la tarea de dibujar una línea polimórfica

2.3.5. Técnicas del modelado de usuarios

Se considera que a fin de lograr un conocimiento más cabal del usuario se combinen las diversas técnicas, teniendo en cuenta el requerimiento específico y presupuesto disponible para cada situación en particular.

1) OBSERVACIÓN DIRECTA

Es el método más confiable y preciso, especialmente si se orienta a la identificación de clases de usuarios y tareas relacionadas. Permite identificar factores críticos por ejemplo, la presión social, que incidirá profundamente en la conducta del usuario cuando el sistema esté funcionando en un contexto determinado.

Sin embargo, a pesar de sus cualidades, es la opción más costosa, ya que para su implementación requiere de observadores experimentados que realicen las observaciones de cada usuario en forma individual. Debido a ello, generalmente es útil cuando un número reducido de observaciones es suficiente para generalizar patrones de conducta o cuando las hipótesis deben ser probadas.

2) ENTREVISTAS

Recaban experiencias personales, opiniones y motivaciones de conducta. Son importantes para detectar conocimiento de procedimientos y problemas en el uso frecuente de herramientas. Implica un costo menor en relación a la observación directa, ya que la codificación de los datos obtenidos es más rápida y fácil de realizar. A pesar de ello, aún se requiere de la figura de un entrevistador experimentado.

3) CUESTIONARIOS

Aportan una visión general de la situación actual y también permiten respuestas acotadas. Pueden incluir tanto preguntas cerradas como abiertas. Permiten un análisis estadístico de los datos, posibilitando así generalizaciones más amplias que las entrevistas. Se recaban gran cantidad de datos a un costo y tiempo sensiblemente menores en relación a las otras técnicas presentadas.

Puede aplicarse bajo distintas modalidades, dirigido (el usuario responde a preguntas realizadas por otra persona que posee el cuestionario); o auto-administrado (el propio usuario lee el cuestionario y responde a las preguntas).

Es el método de menor costo, sobre todo si se aplica bajo la modalidad auto-administrado, ya que no requiere de personal especialmente entrenado y se obtiene gran cantidad de datos a un bajo costo.

4) BAYESIANA

Técnica basada en estudios probabilísticos. El término proviene del teorema de Bayes, que se refiere a la probabilidad de un suceso condicionado por la ocurrencia de otro suceso.

En otras palabras, permite hallar la probabilidad de que una hipótesis sea cierta, una vez comprobado que suceden ciertos eventos, a partir de las probabilidades de que sucedan esos eventos sabiendo que la hipótesis es cierta.

La inferencia bayesiana exige tener a priori una opinión acerca de la proporción p en la población del parámetro a estudiar. Una vez realizada la observación de la proporción p en la población mediante una muestra de sujetos de esta población estudiada se modifica nuestro conocimiento acerca de p , y el mecanismo utilizado para la "actualización" es el teorema de Bayes [Ayçaguer, 2001]. De modo que, una vez aplicado el teorema, toda la información acerca de la proporción desconocida está contenida en la distribución probabilística que se obtiene para p , la llamada distribución a posteriori.

5) BASADAS EN ESTEREOTIPOS

La obtención de información sobre un usuario partiendo de otros es un método simple de adquisición. Se clasifican los usuarios en categorías y se hacen predicciones sobre ellos basándose en un estereotipo asociado a una categoría. Los estereotipos asumen el principio de

que si un usuario pertenece a una categoría, entonces tendrá características y/o comportamientos semejantes a los miembros de esa categoría, bajo un determinado conjunto de circunstancias.

Los componentes principales de un estereotipo son: el cuerpo, que mantiene la información que es “verdadera” para los usuarios a los que se aplica ese estereotipo; y un conjunto de condiciones de activación (*disparadores*) de ese estereotipo para aplicarlas al usuario actual que satisfaga esas condiciones.

Por consiguiente, razonar sobre la base de estereotipos, significa evaluar las reglas de activación con una cierta regularidad, y si alguna condición es satisfecha por el usuario actual, entonces se integran las presunciones correspondientes al estereotipo en el perfil de ese usuario. Las condiciones pueden referirse a datos del usuario (“si el usuario tiene interés en bebés, entonces activar el estereotipo ‘padre’”), a datos de utilización (“si el usuario compra dos libros sobre bebés entonces activar el estereotipo ‘padre’”), o datos ambientales (“si el navegador del usuario es una versión beta, entonces activar el estereotipo ‘mago de los ordenadores’”).

6) REGLAS DE INFERENCIA

Son esquemas para construir inferencias válidas. Estos esquemas establecen relaciones sintácticas entre un conjunto de fórmulas llamados premisas y una aserción llamada conclusión.

Estas relaciones sintácticas son usadas en el proceso de inferencia, por el que se llega a nuevas aserciones verdaderas a partir de otras ya conocidas.

2.3.6. Sistemas y modelos de usuario más Relevantes

1) HOW THE WEST WAS WON

Es un ejemplo de éxito temprano del modelado de usuario. El sistema WEST [Burton-Brown,1982] representa un esfuerzo pionero de explorar temas asociados con el modelado de usuarios. WEST fue un sistema de entrenamiento para un juego llamado ‘*How the West was Won*’ que estaba modelado en ‘Toboganes y Escaleras’ (‘*Chutes and Ladders*’). Los jugadores rotaban tres ruletas y tenían que formar una expresión aritmética a partir de tres números que aparecen en las ruletas usando +, -, *, / y los apropiados paréntesis. Por ejemplo, un jugador que consigue dos, tres y cuatro en las ruletas quizá forme la expresión $(2+3)*4 = 20$ y moverá 20 espacios hacia delante. Si los jugadores llegan a un pueblo (los pueblos aparecen cada 10 espacios), ellos se moverán hacia el siguiente pueblo. Si llegan a un tobogán, se tendrán que

deslizar por él hasta el final. La mejor estrategia es imaginar todos los posibles movimientos y elegir aquel que te ponga lo más lejos posible de tus oponentes. Pero análisis empíricos mostraron que los estudiantes no utilizaban esta estrategia, ellos confiaban más en la estrategia de sumar los dos números menores y multiplicarlos un el mayor.

WEST analizó los movimientos de los estudiantes en términos de la estrategia óptima, tal que no daba oportunidad de llegar a un pueblo, a un tobogán o a un oponente. Si WEST descubriera tal modelo podría intervenir en el momento oportuno, cuando el movimiento del estudiantes estuviera muy lejos de lo óptimo, lo que indicaría al estudiante que podría haberlo hecho mucho mejor. Esto entonces daría al estudiante una oportunidad para ejecutar el movimiento.

En el contexto de WEST, fueron explorados los siguientes problemas del modelado de usuarios:

- Contexto compartido: los ordenadores son restringidos para entrometerse en los defectos de los estudiantes desde lo que hacen en el contexto del juego o solucionar el problema.
- Iniciativa e intrusismo: el modelo de usuario fue usado para efectuar un juicio de cuando dar un consejo valioso y hacer comentarios relevantes a estudiantes sin ser intruso como para destruir la diversión del juego y para evitar el peligro de que los estudiantes nunca desarrollaran las habilidades necesarias para examinar su propio comportamiento y buscar las causas de sus propios errores, porque WEST indica inmediatamente el error del estudiante.
- Relevancia: WEST desarrolló el paradigma ‘entrenar con cuestiones y ejemplos’. Evaluando el contexto y actuando de manera acorde, los estudiantes fueron entrenados en el sentido de que ellos podían ver la utilidad de la cuestión en el momento en que ellos estaban más receptivos a la idea que se les estaba presentando. Basándose en información contenida en el modelo de usuario, el sistema usó una intervención explícita y muestra estrategias para permitir al sistema decir lo correcto en el momento correcto.

A pesar de que el sistema WEST exploró algunos de los conceptos básicos del modelado de usuarios, lo hizo en un contexto de un dominio muy simple donde los resultados eran limitados a la combinación pocas variables. El enfoque es correcto, porque el experto ordenador (como un componente del todo el sistema) hacía operaciones en un mundo ‘cerrado’ puede jugar de manera óptima, y puede determinar el rango completo de alternativas de comportamientos. A

bajo nivel, los sucesos individuales era fáciles de interpretar. El modelo de usuario era construido de forma incremental, explotando muchos sucesos ocurridos en el mismo dominio.

2) INTERBOOK

Es una herramienta para autorizar y enviar libros electrónicos adaptables en la World Wide Web (WWW). Proporciona una tecnología para desarrollar libros electrónicos desde un texto plano hasta textos en lenguaje HTML. También proporciona un servidor HTTP para envíos adaptativos de estos libros electrónicos a través del WWW. Para cada usuario registrado, un servidor de InterBook mantiene un modelo individual de conocimiento del usuario y aplica este modelo para proporcionar guías, soporte de navegación y ayuda adaptativos.

InterBook aplica algunos resultados de investigación en el área de *Adaptive Hypertext and Hypermedia*. Esta investigación demuestra que el soporte de navegación adaptativa puede hacer la navegación hipermedia más productiva y puede proteger a los usuarios de perderse en el hiperespacio. Las técnicas de soporte de navegación adaptativa aplicadas en InterBook son eficientes para aplicaciones educativas de hipertextos e hipermedia.

InterBook es usado para enviar distintos cursos adaptativos basados en la Web.

3) SISTEMA ORIMUHS

Se trata de un sistema de ayuda hipermedia que soporta contextos sensitivos y presentaciones adaptables al usuario de ayuda hipermedia, proporcionando ayuda controlada por el usuario y recuperación de información adicional basada en agentes. Incorpora un modelo de usuario sofisticado con estereotipos (niveles de experticia). Está implementado en imágenes médicas y sistemas CAD.

Ha sido desarrollado conceptualmente e implementado como una extensión de nuevos y existentes interfaces de usuario gráficas.

Construye jerarquías de conceptos de acción en un contexto, donde cada unidad contiene sus sub-unidades, se asigna un LOD a cada unidad y se define un criterio de ‘completo de manera exitosa’ para cada unidad. También define las correspondencias entre el contexto/concepto de trabajo y soporte de usuario. Por último graba las acciones de los usuarios pre-clasificados para construir gráficos de acción.

En la siguiente figura se muestra la arquitectura del sistema:

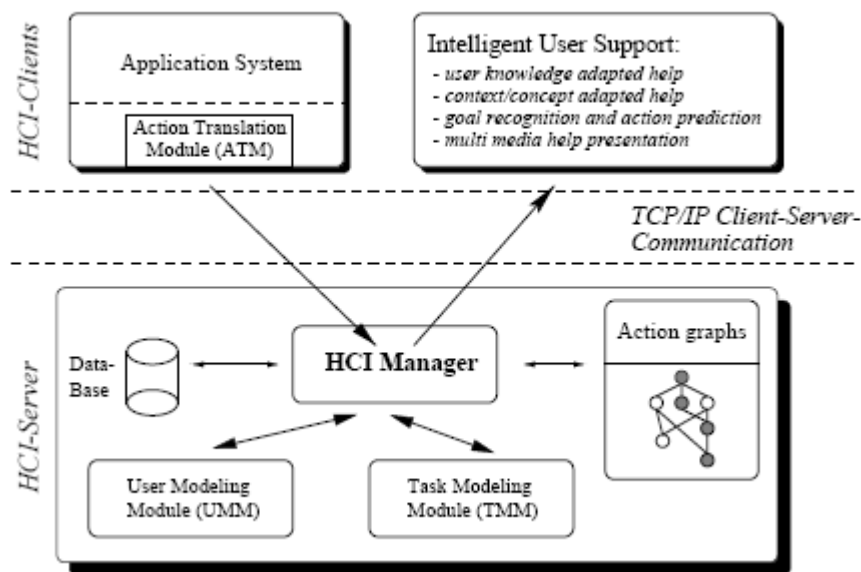


Ilustración 5. Arquitectura del sistema ORIMUHS

4) BGP-MS

Fue desarrollado por Alfred Kobsa y Wolfgang Pohl en 1995 [Kobsa, 1994] . BGP-MS (*Belief Goal and Plan Maintenance System*) es un modelo de usuario shell que puede asistir a sistemas de software interactivo adaptándolo a los usuarios basándose en suposiciones sobre su conocimiento, opiniones y metas. En tiempo de ejecución, cuando el usuario interactúa con la aplicación, BGP-MS acepta información de la aplicación sobre las opiniones observadas, metas y acciones del usuario. Puede obtener información del usuario mediante cuestionarios. La información sobre las opiniones y las metas del usuario que actualmente está comunicado con BGP-MS llega a estar representada en el modelo de usuario individual del propio usuario. Las suposiciones adicionales son adquiridas con varios tipos de inferencias que están basadas en las actuales entradas (de datos) en el modelo de usuario, las contestaciones del usuario en el cuestionario, las acciones del usuario comunicadas por la aplicación del sistema y el conocimiento estereotípico sobre los subgrupos de usuario.

Se trata de un sistema de hipertexto adaptado al usuario-cliente. Puede ser considerado como la "caja negra" que recibe información sobre las cuestiones que pueda plantear un usuario. Es decir, es un sistema que se está continuamente adaptando a las necesidades cambiantes del usuario.

Proporciona un conjunto integrado de mecanismos de representación del conocimiento para representar sus suposiciones sobre el usuario actual, su dominio específico del conocimiento modelado del usuario, y opcionalmente su conocimiento general sobre el dominio de la aplicación. La representación de la capa externa se realiza mediante *particiones*, que pueden ser ordenadas en jerarquías. Esta representación permite mostrar las suposiciones sobre las opiniones y metas del usuario con respecto al dominio de aplicación, el conocimiento del sistema sobre el dominio de aplicación, y las suposiciones concernientes a las características relevantes de subgrupos de usuarios (también llamados *estereotipos*). Las reglas de inferencia que están basadas en opiniones y metas observadas, las respuestas de un usuario en una entrevista, las acciones del usuario, y los contenidos del modelo actual de usuario también pueden ser representadas.

También proporciona algunos métodos de inferencia para derivar suposiciones sobre el usuario basándose en mensajes de la aplicación y en las suposiciones realizadas previamente.

5) SISTEMA ACTIVIST

ACTIVIST [Fischer, 1985] es un sistema de ayuda activa para un editor parecido a EMACS. Apoyó a la gente a incrementar su aprendizaje acerca de las piezas funcionales de relevancia de sus tareas. Dirigió el problema a la gente que a menudo aprende recibiendo respuestas a preguntas que nunca se habían planteado o que no eran capaces de plantearse. ACTIVIST está capacitado para tratar con dos tipos diferentes de comportamiento subóptimo: el usuario que no sabe un comando complejo y usa comandos no óptimos para conseguir el objetivo; y el usuario que sabe el comando complejo que no usa la mínima secuencia clave para emitir el comando.

Similar al observador humano, ACTIVIST estaba preparado para manejar las siguientes tareas: reconocer qué va a hacer el usuario o qué quiere hacer, evaluar cómo el usuario intenta conseguir el objetivo, construir un modelo (acumulando información sobre períodos de uso largos) del usuario basado en los resultados de la tarea de evaluación, y decidir (dependiendo de la información en el modelo) cuándo y cómo interrumpir.

La tarea de reconocimiento y evaluación en ACTIVIST estaba delegada en 20 planes diferentes especializados. Como un ejemplo de sistema de modelado de usuarios, ACTIVIST modeló sus usuarios en la ‘observación actual’ de acciones más que en creencias inferidas [McKeown, 1990] . Contrario a lo que hacía el sistema WEST, ACTIVIST operaba con un

entorno mucho más delimitado, creando el desafío de inferir los objetivos de los usuarios desde las interacciones de bajo nivel. La filosofía básica de diseño de ACTIVIST permite a los usuarios ignorar el aviso proporcionado por el sistema; si ellos ignoran la información proveída por un plan específico especializado repetidamente, dicho plan será desechado por ACTIVIST.

6) LUMIERE

El proyecto Lumiere [Horvitz, 1998] se centra en la probabilidad y la utilidad para proveer asistencia a los usuarios de software. Fue desarrollado por Eric Horvitz en 1993.

La investigación de Microsoft sobre el proyecto Lumiere se ha centrado en métodos para razonar bajo incertidumbre sobre las metas de los usuarios del software. El centro de la investigación de Lumiere y sus prototipos son modelos Bayesianos de usuario que capturan las relaciones inciertas entre las metas y las necesidades de un usuario y las observaciones sobre el estado del programa, secuencia de acciones fuera de tiempo, y palabras en las preguntas de los usuarios.

El proyecto Lumiere se inició en 1993, y el prototipo inicial de Lumiere/Excel fue presentado por la división de Microsoft Office en Junio de 1994. Investigación en nuevas aplicaciones y extensiones continuadas en paralelo con esfuerzo para integrar porciones del prototipo en las aplicaciones comerciales de software disponibles.

Se centra en emparejar la probabilidad con la utilidad para proporcionar ayuda a los usuarios de software de ordenadores. Un prototipo de Lumiere sirvió como base para el asistente de Office en la versión de 1997.

Las metas de alto nivel del proyecto Lumiere representan los aspectos clave del modelado de usuario y la asistencia automática en las aplicaciones de computación. El modelo de decisión considera las metas y las necesidades de los usuarios. Las metas son las tareas o subtarefas objetivo en el centro de la atención del usuario. Las necesidades son información o acciones automatizadas que reducirán el tiempo y el esfuerzo requerido para alcanzar los objetivos.

El sistema Lumiere/Excel

El prototipo Lumiere/Excel fue construido para demostrar a la división de Microsoft Office el potencial de los modelos de usuario Bayesianos. Se construyó un modelo de usuario Bayesiano que razonaba sobre aproximadamente cuarenta problemas abarcando una gran porción de la aplicación Excel.

En el mundo real: el asistente de Office

Como se puede observar a la vista de la Ilustración 6, el equipo de Office dotó al asistente con un frontal basado en caracteres para transmitir los resultados de la inferencia Bayesiana, tal y como se comprometió.

The screenshot shows the Office Assistant interface with a character-based display. The main area contains a grid of data, likely representing the results of a Bayesian inference process. The data is organized into columns and rows, with some cells containing numerical values and others containing text. A small dialog box is visible in the center, asking '¿Qué desea hacer?' (What do you want to do?).

Ilustración 6. Captura de pantalla del asistente de Office

El sistema usa un rico conjunto de variables de contexto que capturan la información sobre la vista actual y los documentos. Sin embargo, el sistema no emplea la información persistente del perfil de usuario, y no razona sobre la competencia. Además, el sistema no usa las combinaciones de eventos sobre el tiempo. Solo se considera un pequeño conjunto de acciones atómicas de usuario.

7) *THEO*

Se trata de un mecanismo de representación e inferencia. Un modelo representado en THEO está compuesto por una serie de entidades, como por ejemplo, la entidad denominada usuario. Cada entidad está compuesta a su vez por una serie de atributos, como por ejemplo nombre o dirección. El valor de estos atributos puede determinarse de diferentes formas:

- El valor del atributo lo proporciona directamente el usuario o el tutor (por ejemplo, el atributo nombre) o bien es el resultado de una determinada función (por ejemplo, la edad del

alumno se calcula a partir de la fecha de nacimiento y la fecha actual en la que ocurre la interacción).

- El valor del atributo se infiere gracias a la aplicación de una determinada regla predefinida por el tutor. La sintaxis de estas reglas, es similar a la de las reglas en el lenguaje de programación PROLOG. Una característica muy importante en THEO es la consistencia, esto es, si una regla depende de un atributo en particular y el valor de dicho atributo varía en una interacción posterior, entonces el valor del atributo asociado mediante esta regla, también cambia.

- El valor del atributo puede ser el resultado de una tarea de aprendizaje automático. Cuando un atributo depende de una tarea de aprendizaje, THEO manda la petición a un módulo encargado de ejecutar dicha tarea. Como veremos en el siguiente apartado, nuestro objetivo es mejorar la eficiencia de dichas tareas. Para ello proponemos la combinación de diferentes técnicas de aprendizaje automático que nos permita soslayar los problemas que en la ejecución de estas tareas suponen la heterogeneidad de los datos (tanto en los valores aprendidos como en los atributos que los representan).

8) SISTEMA PUSH

Todos queremos recibir en nuestros ordenadores la información precisa: ni demasiados datos ni referencias incompletas. El uso masivo de Internet está permitiendo que se generalicen sistemas para el suministro de noticias que funcionan con el tradicional esquema de la "suscripción". Entre todos ellos destacan, por sus grandes posibilidades, los servicios Push.

A la tecnología push se le ha denominado en castellano a veces como "recepción pasiva". En este caso es el servidor el que contacta con el cliente para informarle de la existencia de noticias. Cuando el material está 'preparado' se envía un aviso, generalmente de forma transparente para el suscriptor, a una aplicación cliente.

Obviamente, antes de este paso, existe un proceso previo, mediante el cual el usuario se trae de la Red el software apropiado, lo instala en su máquina y se suscribe a una serie de 'canales'. Estos canales están especializados en temas concretos: deportes, actualidad, pronósticos del tiempo, noticias internacionales...

Se puede observar que existe cierto paralelismo entre la tecnología Push y la televisión, pero las diferencias son notables. Para empezar, hoy por hoy la selección puede ser más 'refinada' usando sistemas Push. Con la televisión tradicional, como mucho, es posible optar

por un canal dedicado a un tema u otro. Si, por ejemplo, escogemos un canal de deportes, posiblemente recibamos información de algunos deportes que no nos llamen para nada la atención. Sin embargo, un canal push permite definir más los intereses de los usuarios, posibilitando la elección de uno o varios deportes.

Por otra parte, la televisión es ‘instantánea’. Con esto queremos decir que la información se comunica (con más o menos rapidez) en un momento determinado. Si entonces se está viendo ese canal se conocerá la noticia. En caso contrario se perderá. Con software que permite la recepción push es posible almacenar durante un tiempo ilimitado la noticia, que podrá ser consultada en cualquier momento. Tampoco es necesario tener el ordenador encendido todo el día, ya que cuando se realiza una conexión, la información se actualiza automáticamente.

Además debemos señalar que las noticias que se presentan en televisión están ‘limitadas’. Tan sólo se obtiene la información que la cadena de televisión estima oportuno. No existe la opción de profundizar más en un tema que interese al usuario. Por contra, todo documento que se transmita por la Red es susceptible de incluir links a otras direcciones, texto, imágenes, vídeo, applets realizadas en diversos lenguajes, sonido y un largo etcétera que permite una mayor interactividad del usuario con la noticia o, incluso, con la fuente de información.

Es la tecnología push la que ofrece un software más inteligente y con mayor capacidad para utilizar Internet como fuente de noticias a la medida y en el momento preciso.

Los programas push tienen diversas formas de presentar la información. El pionero de este tipo de tecnología, PointCast, utiliza un salvapantallas propio, cuyo funcionamiento describiremos a continuación. Éste es el modelo más utilizado, lo que no significa que sea el único. Por norma general, se suele emplear el protector de pantallas como una opción más. BackWeb, por ejemplo, además de este elemento, cuenta con animaciones que surgen dentro de nuestra pantalla, proporcionando acceso directo a la noticia.

Independientemente del procedimiento que utilicen, de momento todos presentan como principal inconveniente el exceso de ‘localismo’ en la información suministrada. Actualmente la mayor parte de los canales ofrece datos sólo sobre Estados Unidos.

GESTIÓN DEL PROYECTO

DE TODO LO VISIBLE E INVISIBLE

3. GESTIÓN DEL PROYECTO

En este capítulo se van a explicar de forma detallada las distintas fases por las que ha ido pasando el proyecto hasta conseguir el modelo final: detallando los objetivos iniciales, los requisitos derivados de éstos, el análisis de viabilidad, las pruebas realizadas al software necesarias para su aceptación y describiendo la metodología utilizada.

Se subdivide en dos grandes apartados, correspondientes a cada una de las aplicaciones realizadas para la consecución del presente proyecto: el Web Crawler, encargado de obtener información para la posterior creación de un modelo de usuarios, y el Entorno de trabajo con modelos de usuario, que proporciona una sencilla pero potente interfaz para poder manejar modelos de usuario de distinta naturaleza.

3.1. WEB CRAWLER

3.1.1. Establecimiento de objetivos

Esta parte del proyecto surge debido a la gran cantidad de datos que requiere un modelo de usuario. Tal y como quedó reflejado en los primeros capítulos, cuanta más información contenga un modelo de usuarios mayor calidad tendrán los datos proporcionados por el mismo.

Además, debido a que los modelos de usuario son un herramienta indispensable en la vida cotidiana de creadores de software, productos cosméticos, marketing, puesto que define a la perfección el público para el cual se está trabajando, obteniendo así cierta ventaja sobre la competencia, presentando productos especializados para cada persona o colectivo, mejor aún si la información contenida en dichos modelos es lo más real posible.

Para obtener información real se pensó en recuperarla de alguna de las redes sociales existentes, ya que resultan ser una fuente inagotable de información acerca de sus usuarios. En gran parte de estas redes la mayoría de la información sobre los usuarios queda restringida sólo a los componentes de dichas redes.

Una vez analizadas varias de las redes sociales, atendiendo a la información contenida en ellas: número de usuarios, información pública disponible, etc. se eligió *Facebook* como fuente de información. Los datos a obtener de cada uno de los perfiles de usuario públicos en esta red

fueron las páginas de las que son admiradores (ya sean alimentos, personas famosas, marcas de ropa, etc.) y el número de amigos que tiene en esa red.

La necesidad de recavar la información de manera rápida y precisa y la gran cantidad de datos contenida en *Facebook* hizo que se precisara una herramienta tal como un crawler para poder extraer y manejar el gran flujo de datos. Además, debido a que el problema radicaba en extraer información de una página Web, se propuso la creación de un Web crawler como solución.

3.1.2. Especificación de requisitos

Para comenzar se listará el catálogo de los requisitos encargados de cubrir la completa funcionalidad de la aplicación requeridas por el usuario final, esto es, los denominados requisitos de usuario.

En la definición de estos requisitos se pretende recoger qué es lo que quiere el cliente que realice la aplicación, identificando las funcionalidades y restricciones. El equipo de desarrollo ha ayudado al cliente a realizar esta definición, además de revisar la lista obtenida.

Existen dos tipos de requisitos de usuario: de capacidad y de restricción.

Después de los requisitos de usuario se presentan los requisitos de software, que son una referencia para verificar el diseño y el producto, pues ambos deben cubrir los requisitos que se hayan marcado. Se dividen en funcionales, de rendimiento, de interfaz, de operación, de recursos de comprobación, de aceptación, de documentación, de seguridad y de calidad.

Se presentan dos tablas resumen con todos los requisitos recogidos, para mayor profundidad en la especificación de los mismos se puede acudir al ANEXO A. Aplicación Web Crawler.

El campo ID hace referencia al identificador del requisito, que debe ser unívoco. El formato a seguir es el siguiente: R<tipo>-<números>. En este caso el campo <tipo> se corresponde con caracteres del nombre de tipo de requisito (las cuales se definen en la Tabla 1) y <números> son números enteros consecutivos incrementados en una unidad.

Tipo de requisito	<tipo>
Capacidad	CP
Restricción	RS
Funcional	FN
Rendimiento	RD
Interfaz	IN
Operación	OP
Recursos	RC
Comprobación	CM
Documentación	DC
Seguridad	SG

Tabla 1. Definición de identificadores de tipos de requisitos

ID	Nombre
RCP-001	Acceso al contenido de una página Web de <i>Facebook</i> e interpretarlo
RCP-002	Distinguir páginas con información relevante de las que sólo tienen enlaces
RCP-003	Conocer las direcciones de las páginas ya procesadas
RCP-004	Obtener información de una página que contenga un perfil de usuario
RCP-005	Almacenar la información obtenida del perfil en una base de datos
RCP-006	Inicio de la aplicación desde un enlace a una página de <i>Facebook</i> pasado como parámetro
RCP-007	Extraer todos los enlaces a otras páginas desde la página procesada
RCP-008	Acceder a todas las posibles páginas de Facebook, a través de los enlaces
RRS-001	Las páginas de Facebook accedidas estarán en castellano

Tabla 2. Especificación de requisitos del Web Crawler (1/3)

ID	Nombre
RRS-002	Tiempo de respuesta de la base de datos no será superior a un segundo
RRS-003	Todas las páginas con información de perfil tienen el mismo formato
RRS-004	Todas las páginas con información no relevante tienen el mismo formato
RRS-005	La base de datos empleada para almacenar la información extraída será Oracle
RRS-006	Las listas de páginas Web visitadas y pendientes de visitar se almacenarán en una base de datos
RRS-007	El contexto de las direcciones de las páginas Web es: http://www.facebook.es/
RRS-008	Las direcciones Web almacenadas no incluirán la parte del contexto
RRS-009	Máximo 10 millones de registros en las tablas que contienen información de perfiles de usuarios
RFN-001	Procesamiento de páginas Web dentro de contexto
RFN-002	Clasificación de las páginas Web procesadas
RFN-003	Extracción de los enlaces contenidos en una página Web
RFN-004	Extracción de información relevante contenida en una página Web
RFN-005	Almacenamiento de la información relevante en base de datos
RFN-006	Ejecución de la aplicación partiendo de una dirección Web pasada como parámetro
RFN-007	Ejecución de la aplicación partiendo de un estado anterior
RFN-008	Mantenimiento de listas con Webs pendientes de visitar y ya visitadas
RFN-009	Almacenamiento de enlaces extraídos de una página
RFN-010	Procesamiento de páginas pendientes de visitar
RRD-001	Aplicación disponible 24 horas al día
RIN-001	Trazas de direcciones de páginas Web procesadas
RIN-002	Mensajes de error

Tabla 3. Especificación de requisitos del Web Crawler (2/3)

ID	Nombre
ROP-001	Inicialización de las listas de Webs visitadas y pendientes de visitar
ROP-002	Contexto por defecto del parámetro de inicio
ROP-003	Añadir un enlace a la lista de enlaces pendientes de visitar si no estaba previamente
ROP-004	Contexto por defecto en enlaces
ROP-005	Contextos válidos de los enlaces
ROP-006	Descarga de contenidos de páginas Web
ROP-007	Añadir contexto a enlace extraído de lista de pendientes de visitar
ROP-008	Descarga de información de una página Web
ROP-009	Descarga de enlaces de una página Web
ROP-010	Procesamiento de enlaces
RRC-001	Tamaño máximo de tablas
RCM-001	Comprobación de parámetros
RCM-002	Nombres de tablas en la base de datos
RDC-001	Lenguaje castellano
RDC-002	Manual de usuario
RDC-003	Comentarios del código fuente
RDC-004	Tipo de letra en la documentación
RDC-005	Tamaño de letra en la documentación
RDC-006	Legibilidad del manual de usuario
RSG-001	Almacenamiento del estado del sistema

Tabla 4. Especificación de requisitos del Web Crawler (3/3)

3.1.3. Análisis de viabilidad

1) SOFTWARE

A continuación se va a evaluar el software necesario para cumplir con los requisitos especificados con anterioridad.

Surge la necesidad de un entorno de ejecución: Java será el elegido, ya que hace viable el objetivo, proporcionando todos procedimientos necesarios para cubrir los requisitos (dispone de bibliotecas para acceder a direcciones Web, o URLs, de manejo de base de datos, etc.).

Además, no será necesaria la utilización de ningún navegador, ya que el acceso a las páginas Web lo proporciona Java.

Por otro lado se tiene la plataforma de ejecución, que no será otra sino Windows, ya que es la plataforma más difundida y es aceptada por todas las versiones de Java y Oracle.

La herramienta de desarrollo del sistema elegida es Eclipse, debido a que el equipo de desarrollo tiene formación y experiencia previa en dicha herramienta, es de distribución gratuita y soporta todas las versiones de Java.

2) HARDWARE

Será necesario disponer de una máquina capaz de procesar las páginas en un tiempo razonable y con conexión a Internet. Por ello, los requerimientos mínimos de la máquina donde se ejecutará la aplicación son:

- Procesador Intel® Core® 2 Quad 2,33GHz
- Tarjeta gráfica de 512 MB
- 4GB de memoria RAM
- Disco duro de 200GB, con dos particiones: una para el SO y otra para los datos
- Conexión de red Ethernet

3.1.4. Pruebas de validación

En este punto se van a enumerar todas las pruebas necesarias para comprobar que el sistema cumple con todos los requisitos que ha marcado el usuario. Son necesarias para que el usuario valide el sistema, antes de la puesta en explotación.

Se debe insistir sobre todo en probar que el sistema cumple los criterios de aceptación del sistema que sirven de base para asegurar que satisface los requisitos exigidos. Estos criterios de aceptación quedan definidos de forma clara prestando atención a: procesos identificados como críticos, rendimiento, seguridad y disponibilidad del sistema. Los puntos de atención dependen de la naturaleza del sistema y los requisitos marcados por el usuario. Para consultar con más detalle las pruebas se puede acudir al ANEXO A. Aplicación Web Crawler.

ID	Nombre
PR-001	Descargar el contenido de una página Web
PR-002	Cambio de contexto correcto
PR-003	Clasificar una página por su contenido
PR-004	Obtener información relevante de una página
PR-005	Obtener los enlaces de una página
PR-006	No procesar una página Web ya procesada
PR-007	Procesar todos los enlaces de una página
PR-008	Almacenar los datos obtenidos de una página Web con información relevante
PR-009	Acceso a una página Web fuera del contexto de la aplicación

Tabla 5. Especificación de pruebas del Web Crawler

3.1.5. Metodología del desarrollo

Siguiendo una metodología estructurada de desarrollo de software, se han seguido una serie de pasos, en un orden determinado, para llevar a cabo la construcción exitosa del sistema. A continuación se presenta una breve descripción de las distintas fases:

- Estudio de las ideas iniciales acerca del proyecto.
- Análisis el sistema: plasmar las ideas iniciales, definiendo la funcionalidad del sistema completo.
- Diseño del sistema: determinación del sistema definitivo, a partir del análisis definido previamente.
- Codificación: implementación de todo lo anterior.

Para la consecución del proyecto se ha utilizado un ciclo de vida en espiral (Ilustración 7). La elección de este ciclo de vida se debe a que el proyecto está completamente identificado con las características de dicho modelo; por ejemplo, las incertidumbres sobre los resultados realmente alcanzables y/o las ignorancias sobre el comportamiento de las tecnologías, impiden realizar un proyecto con especificaciones cerradas.

Se parte de la definición de unas especificaciones iniciales para hacer un prototipo (es decir, un producto parcial) y provisional. El propio producto a lo largo de su desarrollo puede considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

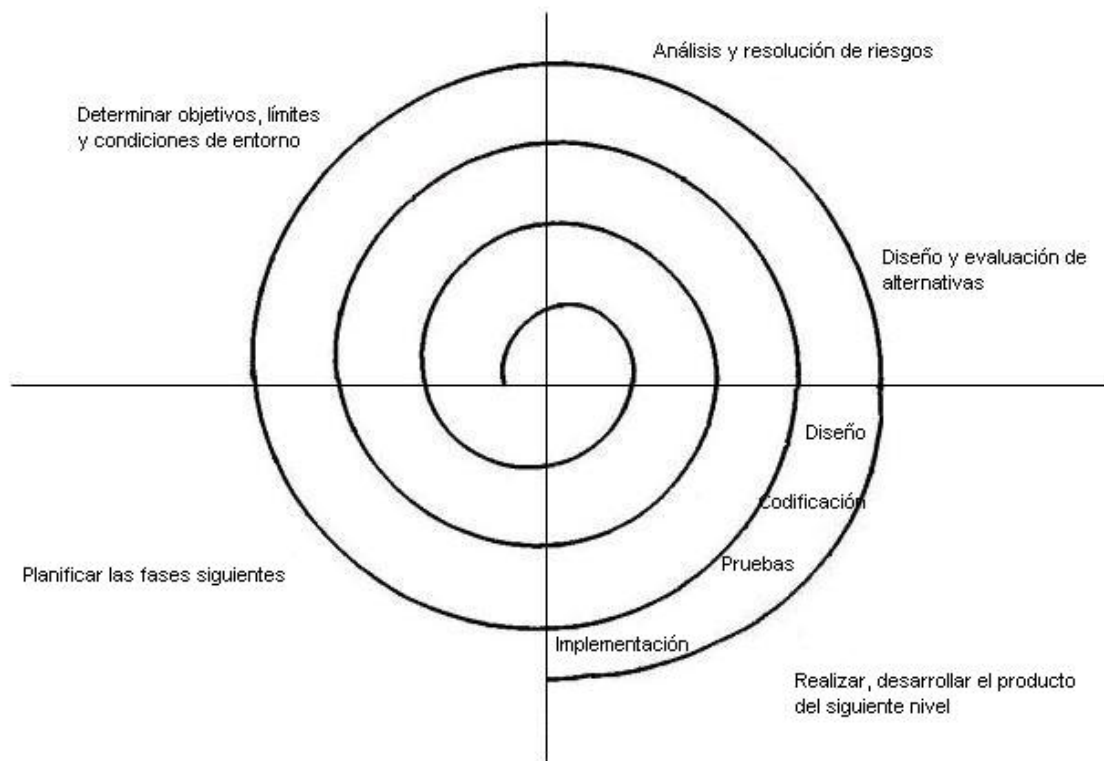


Ilustración 7. Ciclo de vida en espiral

3.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO

3.2.1. Establecimiento de objetivos

Los modelos de usuario son una herramienta indispensable en la vida cotidiana de creadores de software, productos cosméticos, marketing, ya que define a la perfección al público para el cual se está trabajando, obteniendo así cierta ventaja sobre la competencia, presentando productos especializados para cada persona o colectivo.

Esta parte del proyecto surge debido a la gran cantidad de datos que requiere un modelo de usuario. Tal y como quedó reflejado en los primeros capítulos, cuanto más información contenga un modelo de usuarios mayor calidad tendrán los datos proporcionados por el mismo.

Por todo ello se ha pensado en realizar un programa que gestione modelos de usuarios, con una interfaz sencilla y que sea capaz de interactuar con el modelo de usuarios elegido.

Partiendo de la comunicación hombre-máquina, se ha creado un proyecto orientado a la gestión de modelos de usuario. Debido a que la máquina necesita conocer al usuario, y las características de cada usuario son muy volubles en cada instante del tiempo para cada aplicación determinada resultará muy útil poder conocer aproximadamente las preferencias y características del usuario correspondiente. Así pues, se comenzó a desarrollar la posibilidad de diseñar e implementar el programa que fuera capaz de:

- Adquirir la información introducida por el usuario.
- Proporcionar la información a través de la inferencia del modelo.
- Comprobar la certeza sobre la información del usuario actual.
- Cerrar la sesión del usuario, de manera que se almacene en el repositorio la información inferida.
- Cancelar la sesión del usuario, sin tener en cuenta la información inferida en dicha sesión.

A parte las ideas iniciales con respecto a la funcionalidad del programa también se estuvieron considerando las diferentes formas de construir el programa: estructura, interfaz, lenguajes, etc.

3.2.2. Especificación de requisitos

Para comenzar se va a proceder a describir el catálogo de los requisitos encargados de cubrir la funcionalidad de la aplicación, requeridos por el usuario final, esto es, los denominados requisitos de usuario.

En la definición de estos requisitos se pretende recoger qué es lo que quiere el cliente que realice la aplicación, identificando las funcionalidades y restricciones. El equipo de desarrollo ha ayudado al cliente a realizar esta definición, además de revisar la lista obtenida.

Existen dos tipos de requisitos de usuario: de capacidad y de restricción.

Después de los requisitos de usuario se presentan los requisitos de software, que son una referencia para verificar el diseño y el producto, pues ambos deben cubrir los requisitos que se

hayan marcado. Se dividen en funcionales, de rendimiento, de interfaz, de operación, de recursos de comprobación, de aceptación, de documentación, de seguridad y de calidad.

Se presentan dos tablas resumen con todos los requisitos recogidos, para mayor profundidad en la especificación de los mismos se puede acudir al ANEXO B. Aplicación Entorno de trabajo con modelos de usuario.

El campo ID hace referencia al identificador del requisito, que debe ser unívoco. El formato a seguir es el siguiente: R<tipo>-<números>. En este caso el campo <tipo> se corresponde con caracteres del nombre de tipo de requisito (las cuales se definen en la Tabla 6) y <números> son números enteros consecutivos incrementados en una unidad.

Tipo de requisito	<tipo>
Capacidad	CP
Restricción	RS
Funcional	FN
Rendimiento	RD
Interfaz	IN
Operación	OP
Recursos	RC
Comprobación	CM
Documentación	DC
Seguridad	SG

Tabla 6. Definición de identificadores de tipos de requisitos

ID	Nombre
RCP-001	Selección del tipo de modelo de usuarios
RCP -002	Cambios en las propiedades de modelo de usuarios elegido
RCP -003	Carga de datos desde el fichero de configuración
RCP -004	Salvar la configuración en un fichero
RCP -005	Adquirir información acerca del usuario actual
RCP -006	Obtener información acerca del usuario actual a través de la inferencia del modelo
RCP -007	Almacenar los datos del usuario actual inferidos sobre el modelo, positivamente
RCP -008	Almacenar los datos del usuario actual inferidos sobre el modelo, negativamente
RCP -009	Cerrar la sesión del usuario actual e iniciarla con un nuevo usuario
RCP -010	Introducir nuevos datos en el modelo a partir de una tabla de una base de datos
RCP -011	Modificar los atributos de la base de datos de la que se extrae la nueva información para introducirla
RCP -012	Evaluar las características de usuarios no existentes en la base de datos
RCP -013	Evaluar las características de usuarios existentes en la base de datos
RCP -014	Modificar los atributos de la base de datos desde la que se va a producir la evaluación de los usuarios
RRS-001	La base de datos empleada para almacenar la información extraída será Oracle
RRS-002	Los tipos de modelos de usuarios son Oracle o Java
RRS-003	Sólo se pueden inferir características que se encuentren introducidas en la base de datos elegida
RRS -004	Sólo se pueden inferir valores que se encuentren introducidas en la base de datos elegida

Tabla 7. Especificación de requisitos del entorno de trabajo con modelos de usuario (1/4)

ID	Nombre
RRS -005	Al seleccionar el rango de valores de identificadores de usuarios, el límite inferior siempre será menor que el límite superior
RRS -006	La evaluación se efectuará sobre, al menos, un usuario
RRS -007	La adquisición de información inferida sólo podrá efectuarse en el caso de que se hayan inferido un átomo o varios de información
RRS -008	La aplicación estará implementada en castellano
RFN -001	Selección de un modelo de usuario de tipo Java
RFN -002	Selección de un modelo de usuario de tipo Oracle
RFN -003	Lectura del fichero de configuración
RFN -004	Cambiar el modelo de usuarios seleccionado
RFN -005	Obtener información sobre el usuario actual
RFN -006	Inferir información sobre el usuario actual
RFN -007	Dada una cierta información sobre el usuario actual, obtener una certeza
RFN -008	Adquirir información del usuario actual positivamente
RFN -009	Adquirir información del usuario actual negativamente
RFN -010	Cerrar la sesión del usuario actual
RFN -011	Seleccionar toda la información de una tabla de una base de datos
RFN -012	Seleccionar un rango de información de una tabla de una base de datos
RFN -013	Almacenar la información obtenida de la base de datos en el modelo de usuarios seleccionado
RFN -014	Cambios en los atributos de la base de datos de la que se extrae información
RFN -015	Evaluación de usuarios no existentes en la base de datos
RFN -016	Evaluación de usuarios existentes en la base de datos
RFN -017	Cambios en los atributos de la base de datos de la evaluación

Tabla 8. Especificación de requisitos del entorno de trabajo con modelos de usuario (2/4)

ID	Nombre
RIN -001	Barra de menú
RIN -002	Cerrar sesión en el menú
RIN -003	Carga del fichero de configuración en el menú
RIN -004	Pestaña Adquisición
RIN -005	Pestaña Consulta
RIN -006	Pestaña Entrenamiento
RIN -007	Pestaña Definición
RIN -008	Pestaña Evaluación
RIN -009	Selección de características y valores
RIN -010	Inserción de información del usuario actual
RIN -011	Lista de inferencias y certezas
RIN -012	Información del modelo de usuarios
RIN -013	Botón Reiniciar
RIN -014	Guardar configuración del modelo en fichero
RIN -015	Selección del rango de usuarios a entrenar
RIN -016	Inserción de las inferencias
ROP -001	Implementación Java del modelo de usuarios
ROP -002	Ejecución de métodos con un modelo de usuarios Java
ROP -003	Ejecución de métodos con un modelo de usuarios Java
ROP -004	Cierre de sesión aprendiendo
ROP -005	Evaluación de usuarios
ROP -006	Característica a inferir en la evaluación

Tabla 9. Especificación de requisitos del entorno de trabajo con modelos de usuario (3/4)

ID	Nombre
RRC -001	Número de valores por característica
RRC -002	Obtención de características de un usuario
RCM -001	Contenido del fichero de configuración
RCM -002	Campos de las tablas de entrenamiento
RCM -003	Parámetros método de adquisición de un átomo de información
RCM -004	Parámetros método de inferencia de un átomo de información
RCM -005	Parámetros método de inferencia de varios átomos de información
RCM -006	Parámetros método de comprobación de un átomo de información
RCM -007	Parámetros método de cierre de sesión
RDC-001	Lenguaje castellano
RDC-002	Manual de usuario
RDC-003	Comentarios del código fuente
RDC-004	Tipo de letra en la documentación
RDC-005	Tamaño de letra en la documentación
RDC-006	Legibilidad del manual de usuario
RSG-001	Ejecución de la aplicación

Tabla 10. Especificación de requisitos del entorno de trabajo con modelos de usuario (4/4)

3.2.3. Análisis de viabilidad

1) SOFTWARE

A continuación se va a evaluar el software necesario para cumplir con los requisitos especificados con anterioridad.

Surge la necesidad de un entorno de ejecución: Java será el elegido, ya que hace viable el objetivo, proporcionando todos procedimientos necesarios para cubrir los requisitos (dispone de bibliotecas para acceder a base de datos, etc.).

Por otro lado se tiene la plataforma de ejecución, que no será otra sino Windows, ya que es la plataforma más difundida y es aceptada por todas las versiones de Java y Oracle.

La herramienta de desarrollo del sistema elegida es Eclipse, debido a que el equipo de desarrollo tiene formación y experiencia previa en dicha herramienta, es de distribución gratuita y soporta todas las versiones de Java, además de proporcionar útiles herramientas para la creación de interfaces gráficas.

2) HARDWARE

Será necesario disponer de una máquina con los siguientes requerimientos mínimos:

- Procesador Intel® Core® 2 Quad 2,33GHz
- Tarjeta gráfica de 512 MB
- 4GB de memoria RAM
- Disco duro de 200GB, con dos particiones: una para el SO y otra para los datos

3.2.4. Pruebas de validación

En este punto se van a enumerar todas las pruebas necesarias para comprobar que el sistema cumple con todos los requisitos que ha marcado el usuario. Son necesarias para que el usuario valide el sistema, antes de la puesta en explotación.

Se debe insistir sobre todo en probar que el sistema cumple los criterios de aceptación del sistema que sirven de base para asegurar que satisface los requisitos exigidos. Estos criterios de aceptación quedan definidos de forma clara prestando atención a: procesos identificados como críticos, rendimiento, seguridad y disponibilidad del sistema. Los puntos de atención dependen de la naturaleza del sistema y los requisitos marcados por el usuario. Para consultar con más detalle las pruebas se puede acudir al ANEXO B. Aplicación Entorno de trabajo con modelos de usuario.

ID	Nombre
PR-001	Obtención de un átomo de información del usuario actual
PR-002	Guardar la información obtenida del usuario actual
PR-003	Inferir un átomo de información del usuario actual
PR-004	Inferir varios átomos de información del usuario actual
PR-005	Comprobar un átomo de información del usuario actual
PR-006	Almacenar datos inferidos
PR-007	Aprender usuario y cerrar sesión
PR-008	Entrenamiento del modelo con todos los registros
PR-009	Entrenamiento del modelo con los registros seleccionados
PR-010	Llamadas a los métodos del modelo
PR-011	Evaluación de usuarios dentro del corpus
PR-012	Evaluación de usuarios fuera del corpus
PR-013	Evaluar varios usuarios
PR-014	Carga de la configuración
PR-015	Guardar la configuración

Tabla 11. Especificación de pruebas del Entorno de trabajo con modelos de usuario

3.2.5. Metodología del desarrollo

Al igual que en la primera parte, se ha utilizado una metodología estructurada de desarrollo de software, siguiendo una serie de pasos, en un orden determinado, para llevar a cabo la construcción exitosa del sistema. A continuación se presenta una breve descripción de las distintas fases:

- Estudio de las ideas iniciales acerca del proyecto.
- Análisis el sistema: plasmar las ideas iniciales, definiendo la funcionalidad del sistema completo.

- Diseño del sistema: determinación del sistema definitivo, a partir del análisis definido previamente.
- Codificación: implementación de todo lo anterior.

De la misma manera que en la primera parte del proyecto, para la consecución del proyecto se ha utilizado un ciclo de vida en espiral (Ilustración 7). La elección de este ciclo de vida se debe a que el proyecto está completamente identificado con las características de dicho modelo; por ejemplo, las incertidumbres sobre los resultados realmente alcanzables y/o las ignorancias sobre el comportamiento de las tecnologías, impiden realizar un proyecto con especificaciones cerradas.

Se parte de la definición de unas especificaciones iniciales para hacer un prototipo (es decir, un producto parcial) y provisional. El propio producto a lo largo de su desarrollo puede considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

ANÁLISIS Y DISEÑO DEL SISTEMA

¿ Qué es lo que quiere ?

¿ Cómo lo quiere ?

4. ANÁLISIS Y DISEÑO DEL SISTEMA

En este punto serán considerados los primeros pasos descritos en el capítulo anterior, efectuando un estudio sobre los mismos para fijar la estructura definitiva del sistema en cuestión.

Una vez examinados con detenimiento las primeras ideas acerca del sistema y su posible entorno se pasará a dar forma a todo lo que el sistema debe ser capaz de hacer, los problemas que se pueden presentar a la hora de plantear todas las tareas que se han de llevar a cabo y las soluciones a estos problemas.

Como se comentó en el capítulo anterior, el proyecto se realiza sobre un ciclo de vida en espiral, lo que permite comprobar constantemente las fases anteriores. Como la etapa de análisis no es una excepción, será revisada ciclo tras ciclo, descubriendo así nuevas funcionalidades y posibilidades que en ciclos anteriores no parecían necesarias y, por tanto, no estaban incluidas en dicho análisis hasta ese momento.

Es en la etapa de análisis donde se deben prever todas las posibles situaciones que se darán más tarde en la etapa de diseño, para no provocar retrasos y cambios bruscos en dicho diseño, ya que ello conllevaría un diseño del sistema equivocado (que no se corresponda con las necesidades descritas) o incluso un posible fracaso del proyecto.

A pesar de que la fase de análisis pueda llegar a parecer superflua, ya que no produce resultados tangibles, resulta imprescindible para la correcta consecución de dicho proyecto. Además, permite al programador tener una idea clara de la funcionalidad que se persigue, así como posibles problemas y/o dificultades que podrían permanecer ocultos si se practica la codificación sin análisis previo.

Seguidamente se podrá comprobar el diseño detallado del proyecto. Se trata de una fase importante, que parte del análisis previo para determinar cómo será el sistema definitivo. Se detallará la elección de la interfaz definitiva del programa, el comportamiento interno y la base de datos que se utilizará para almacenar la información manejada por el sistema.

Una vez analizados todos los requerimientos del sistema en la fase anterior es necesario tomar una decisión definitiva sobre todos los problemas, de encontrar la solución que más se ajuste a nuestro proyecto, para permitir una codificación posterior lo más rápida y consistente posible, minimizando los errores.

Como se viene comentando, no se trata de encontrar la mejor solución en general, ni de encontrar una buena solución a los problemas propuestos, sino de encontrar la mejor solución para este caso determinado. Dicha solución será obtenida gracias al completo conocimiento de los requerimientos y problemas planteados.

Es importante destacar que estas dos etapas forman la piedra angular del proyecto.

Se subdivide en dos grandes apartados, correspondientes a cada una de las aplicaciones realizadas para la consecución del presente proyecto: el Web Crawler y el Entorno de trabajo con modelos de usuario.

4.1. WEB CRAWLER

Es la parte del proyecto encargada de obtener información para la posterior creación de un modelo de usuarios.

4.1.1. Análisis funcional

Teniendo como base lo expuesto en el capítulo anterior, en el que se describía cómo debía ser el sistema, este apartado desarrolla la funcionalidad final del sistema, realizándose mediante la evolución lógica de las ideas surgidas en la funcionalidad inicial.

Primeramente se va a proceder a la identificación de los subsistemas de análisis, de manera que se descompone el sistema de información en diferentes subsistemas de menor tamaño para facilitar la tarea de análisis.

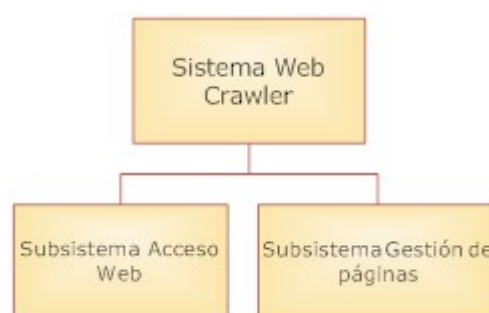


Ilustración 8. Subsistemas de análisis del Web Crawler

A continuación se presenta una descripción de cada uno de ellos:

1) ACCESO WEB

Este subsistema se encarga acceder a las páginas Web, obtenidas a través de direcciones Web contenidas en otras páginas Web o bien de una lista de direcciones Web. Por ello, tiene dos cometidos fundamentales:

- **Gestión de listas de enlaces:** realiza el mantenimiento de dos listas que contienen direcciones Web: una con direcciones pendientes de visitar y otra con las ya visitadas.
- **Acceso a enlaces:** se encarga de acceder a todos los enlaces disponibles en la lista de direcciones Web no visitadas.
- **Contexto correcto:** analiza el contexto del enlace obtenido y los normaliza todos al contexto definido por defecto.

2) GESTIÓN DE PÁGINAS

El cometido de este segundo subsistema es descargar el contenido de las páginas Web, e ir las clasificando y procesando según el mismo. Entonces los objetivos del subsistema son:

- **Descarga de contenido:** una vez accedida a la página, se obtiene el contenido íntegro de la misma para poder proceder a su posterior análisis.
- **Obtención de información:** al estar procesando una página que contenga un perfil de usuario, extrae la información definida como relevante para el presente proyecto.
- **Obtención de enlaces:** dada una página que contiene enlaces a otras páginas Web, extrae todas las direcciones de los enlaces, para que luego puedan ser gestionados por el primer subsistema comentado, el de acceso Web.

Una vez definidos los subsistemas de análisis se van a enumerar los casos de uso del sistema Web Crawler. Para un mayor detalle de los mismos se puede acudir al ANEXO A. Aplicación Web Crawler.

ID	Nombre
CU-001	Descarga del contenido de una página
CU -002	Ir al enlace siguiente
CU -003	Obtener información de la página Web
CU -004	Almacenar información de una página Web
CU -005	Obtener enlaces de una página Web
CU -006	Cambiar de contexto una página Web
CU -007	Almacenar enlaces
CU -008	Eliminar enlaces

Tabla 12. Casos de uso del Web Crawler

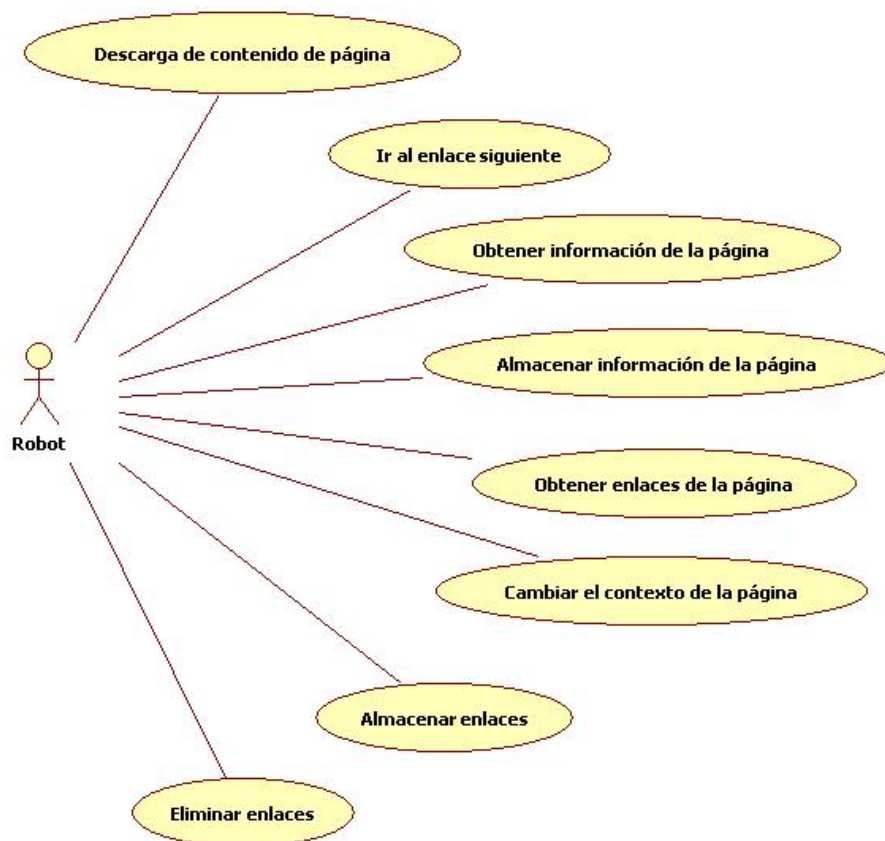


Ilustración 9. Diagrama de casos de uso del Web Crawler

Además, se presentarán los diagramas de secuencia para cada uno de los casos de uso definidos, para ofrecer una visión de cómo será la sucesión de acciones para lograr el objetivo del caso de uso.

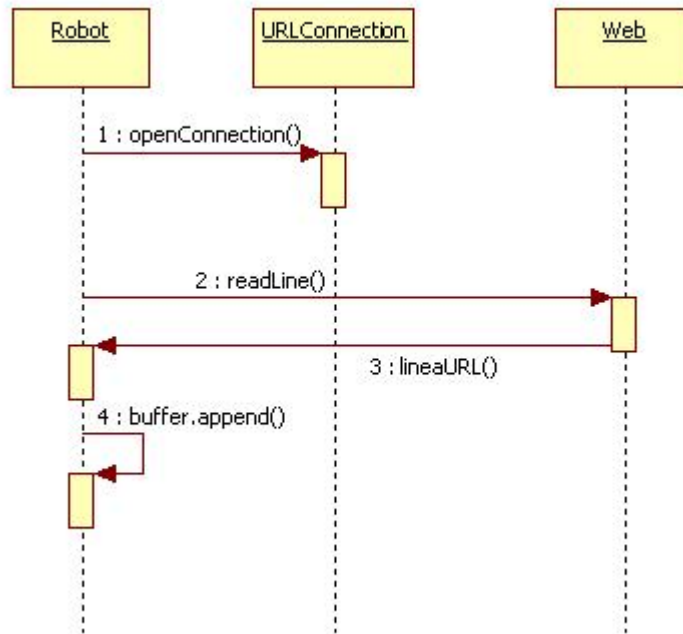


Ilustración 10. Diagrama de secuencia: Descarga del contenido de una página

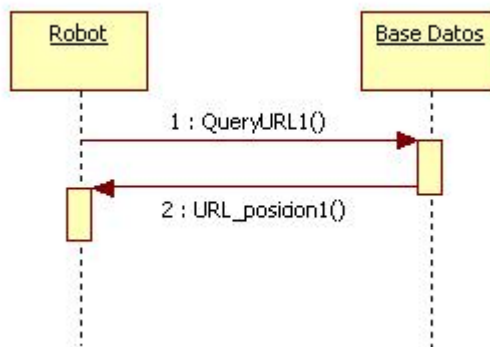


Ilustración 11. Diagrama de secuencia: Ir al enlace siguiente

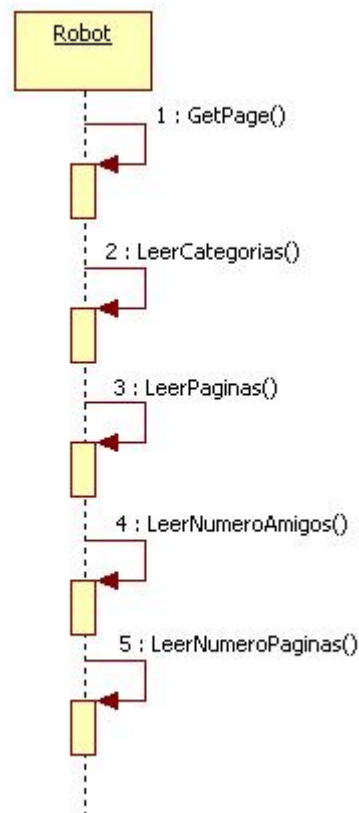


Ilustración 12. Diagrama de secuencia: Obtener información de la página Web

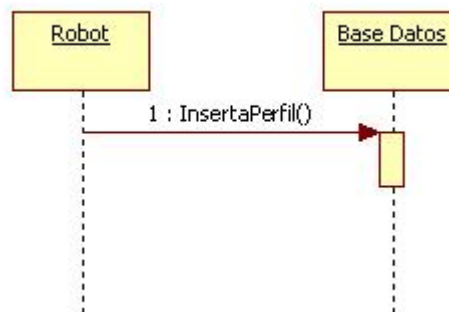


Ilustración 13. Diagrama de secuencia: Almacenar información de una página Web

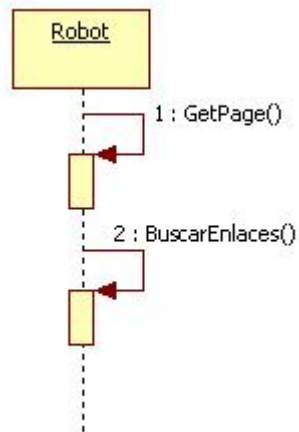


Ilustración 14. Diagrama de secuencia: Obtener enlaces de una página Web

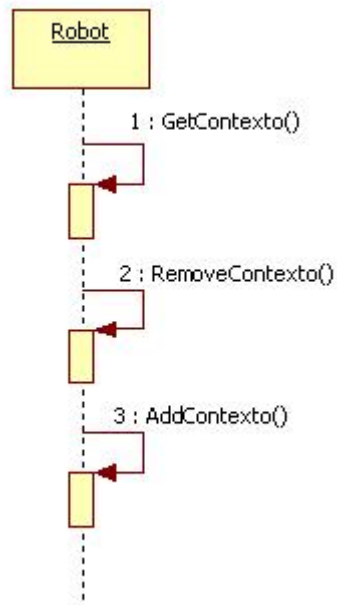


Ilustración 15. Diagrama de secuencia: Cambiar de contexto una página Web

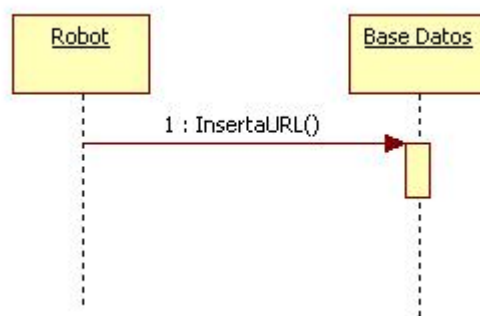


Ilustración 16. Diagrama de secuencia: Almacenar enlaces

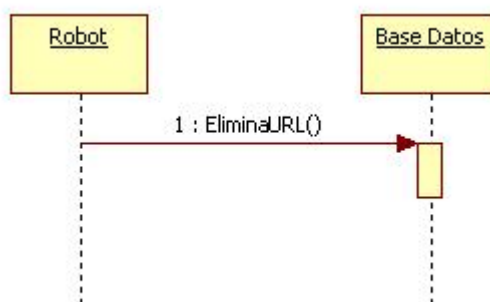


Ilustración 17. Diagrama de secuencia: Eliminar enlaces

4.1.2. Necesidades de información

En este apartado se va a mostrar toda la información relevante al modelo conceptual de datos, descrito mediante un diagrama entidad-interrelación. A través de un estudio detallado se explicarán todas las interrelaciones posibles que se pueden llegar a dar entre cada una de las entidades existentes, las cuales también serán razonadas e ilustradas.

Como se viene comentando, no se trata de encontrar la mejor solución en general, ni de encontrar una buena solución a los problemas propuestos, sino de encontrar la mejor solución para este caso determinado. Dicha solución será obtenida gracias al completo conocimiento de los requerimientos y problemas planteados.

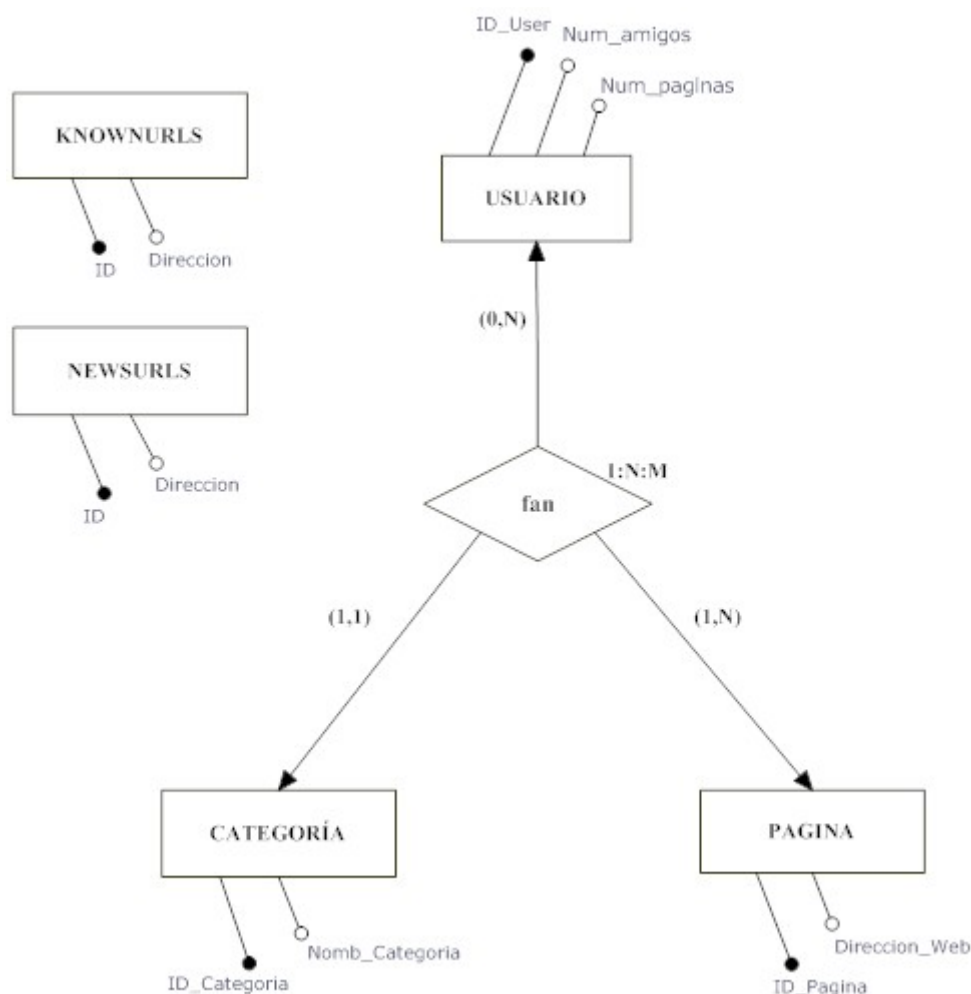


Ilustración 18. Modelo E/R Extendido

En este apartado se explicará con todo detalle las tres entidades que componen el esquema E/R comentado con anterioridad (Ilustración 18):

- **USUARIO:** guarda los datos correspondientes un usuario, extraídos de la página Web de su perfil en *Facebook*. Además del identificador correspondiente al usuario se almacenará el número de páginas de las que es admirador y el número de amigos que tiene.
- **CATEGORÍA:** contiene una lista de todas las categorías en las que se puede clasificar una página Web de la que un usuario es admirador. Se puede referir a cualquier tipo de categoría de las almacenadas en *Facebook*, desde marcas de ropa hasta actores, pasando por música.

- **PÁGINA:** encierra todas las direcciones Web extraídas del perfil del usuario de la página correspondiente de *Facebook* y define los posibles valores para las distintas categorías definidas en la tabla “CATEGORÍA”.
- **NEWSURLS:** contiene todas las direcciones Web extraídas de la página correspondiente de *Facebook* que están pendientes de visitar, es decir, que aún no han sido escaneadas.
- **KNOWNURLS:** describe todas las direcciones Web extraídas de la página correspondiente de *Facebook* que ya han sido tratadas por la aplicación y, por tanto, conocidas; de tal forma que si se va a escanear una página primero se consulta esta tabla para comprobar que no ha sido tratada previamente.

1) INTERRELACIONES

Como se puede observar a la vista del modelo entidad-relación extendido (Ilustración 18) la base de datos está compuesta por una única interrelación: ternaria (‘Fan’, que relaciona las tres tablas existentes), detallada a continuación:

En el modelo definido se trata de una interrelación 1:N:M, pero, en general, esta interrelación ternaria debería ser N:M:P, ya que en dominios más amplios una dirección Web podría pertenecer a más de una categoría.

Al tratarse de una relación ternaria se estudiarán por separado cada una de las entidades participantes desde el punto de vista de la relación con el resto de entidades integrantes.

Para interpretar cómo interviene cada entidad en la relación se utiliza la técnica del 2-1, que consiste en dejar fijas dos entidades mientras se efectúan todas las posibles combinaciones con la entidad restante. A continuación se describen las tres posibilidades existentes:

- **USUARIO – CATEGORÍA / PÁGINA:** Dejando fijas las entidades de “USUARIO” y “CATEGORÍA”, se probarán todas las combinaciones posibles de “PÁGINA”.

Un usuario determinado en una categoría se podrá relacionar, como mínimo, con una tupla de la entidad “página”; ya que si un usuario tiene asignada una categoría determinada tendrá que estar asociado al menos a una página, porque sino no existiría esa relación de “Fan”. Un usuario puede llegar a tener todas las categorías definidas en

el sistema, pero puede no tener almacenadas páginas sobre todas ellas, y en la relación sólo se almacenarán las categorías con su página determinada asociadas a un usuario.

- **USUARIO – PÁGINA / CATEGORÍA:** Dejando fijas las entidades de “USUARIO” y “PÁGINA”, se probarán todas las combinaciones posibles de “CATEGORÍA”.

Para un determinado usuario con una única página, puede haber una y sólo una característica, puesto que en el presente dominio una página determinada sólo puede estar referida a una categoría.

- **CATEGORÍA – PÁGINA / USUARIO:** Dejando fijas las entidades de “CATEGORÍA” y “PÁGINA”, se probarán todas las combinaciones posibles de “USUARIO”.

Para una categoría determinada y una página para la misma, puede referirse a ningún usuario, ya que la categoría con sus valores correspondientes puede estar dada de alta en el sistema, pero que aún no haya usuarios que posean dicha categoría y alguna página de la misma. Por otro lado, como máximo puede estar relacionado con N usuarios, pudiendo existir tuplas de categoría –valor iguales para distintos valores de usuarios.

2) SEMÁNTICA NO CONTEMPLADA

En el modelo entidad / relación extendido, presentado en la Ilustración 18, no se recoge la siguiente semántica:

- Las categorías y las páginas presentan una relación entre ellas, al margen de la relación “Fan”, y es que una categoría puede contener de 1 hasta N páginas. No se representa en el modelo, ya que no influye en el desarrollo del proyecto.
- En el dominio del sistema un usuario asociado a una categoría no va a presentar más de 5 páginas Web por categoría, ya que ésta es la cota máxima de información presentada en el perfil del usuario.
- Un usuario sólo tendrá asociadas 4 categorías como máximo, ya que, una vez más, en el perfil de usuario no aparece más información.

4.1.3. Interfaz

Se proporciona una API en Java con los siguientes puntos de entrada:

- **Arrancar:** se pasará al programa una dirección Web desde la cual el robot pueda iniciar la búsqueda. El sistema no dispone de información previa que procesar, es decir, no existen enlaces pendientes de visitar ni enlaces ya visitados.
- **Re-arrancar:** el sistema no recibe ningún parámetro adicional, ya que obtendrá la información necesaria para el comienzo de la ejecución se encuentra almacenada en la base de datos. Así, la dirección Web desde la cual inicia la búsqueda es la que se encuentre en primera posición en la lista de enlaces pendientes de visitar.

El programa proporciona una traza con las direcciones Web de las diferentes páginas que va procesando. Además, en caso de fallo muestra el mensaje de error correspondiente y continúa con la ejecución.

4.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO

Parte del proyecto que proporciona una sencilla pero potente interfaz para poder manejar modelos de usuario de distinta naturaleza.

4.2.1. Análisis funcional

Para comenzar se va a proceder a la identificación de los subsistemas de análisis, de manera que se descompone el sistema de información en diferentes subsistemas de menor tamaño para facilitar la tarea de análisis.

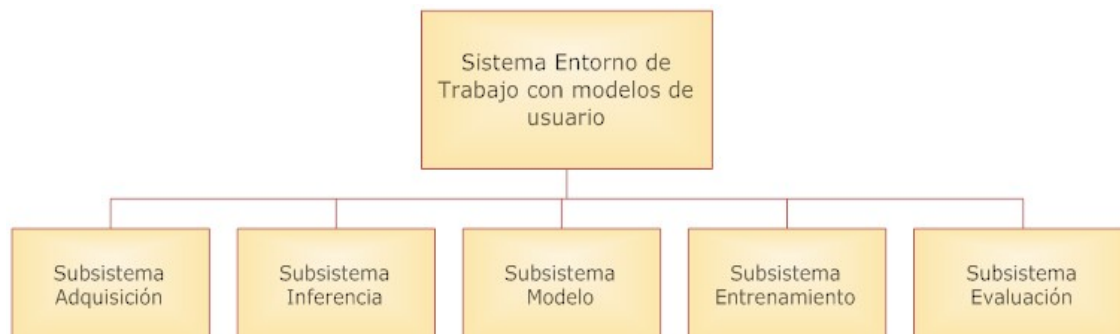


Ilustración 19. Subsistemas de análisis del Entorno de trabajo con modelos de usuario

A continuación se presenta una descripción de cada uno de ellos:

1) ADQUISICIÓN

Este subsistema se encarga de obtener nuevos datos del usuario actual, a través de la interfaz proporcionada por el sistema. Tiene como objetivos principales:

- **Introducir nueva información:** carga en el sistema nuevos pares de característica-valor del usuario actual.
- **Eliminar información:** elimina de la aplicación pares de característica-valor seleccionados del usuario actual.
- **Aprender nuevos datos:** se encarga de almacenar los nuevos datos sobre el usuario actual introducidos en el sistema en la base de datos del modelo.

2) INFERENCIA

Subsistema encargado de realizar consultas sobre el modelo, obteniendo nuevos datos sobre el usuario actual, infiriendo sobre dicho modelo. Los propósitos principales son:

- **Inferir un átomo de información:** obtiene un valor con una certeza para una característica dada sobre el usuario actual, teniendo en cuenta la información ya conocida sobre dicho usuario y la almacenada en el modelo.
- **Inferir varios átomos de información:** obtiene una lista de valores con sus respectivas certezas para una característica dada sobre el usuario actual, teniendo en cuenta la información ya conocida sobre dicho usuario y la almacenada en el modelo.
- **Comprobar un átomo de información:** obtiene una certeza para un valor de una característica dada sobre el usuario actual, teniendo en cuenta la información ya conocida sobre dicho usuario y la almacenada en el modelo.
- **Aprender la información inferida:** se encarga de almacenar la información inferida sobre el usuario actual en el modelo.

3) MODELO

Subsistema cuyo cometido es definir los datos de configuración del modelo de usuarios a utilizar: desde el tipo de modelo hasta el nombre de los métodos del mismo. Los objetivos de este subsistema son:

- **Obtención de información:** los datos de configuración del modelo se pueden introducir a través de la interfaz proporcionada por la aplicación o bien realizando la carga de un fichero de configuración que contenga los datos acerca de la misma.
- **Almacenamiento de información:** los datos recogidos en la interfaz del sistema también pueden ser guardados en un fichero de configuración, para poder proceder a su posterior carga.

4) ENTRENAMIENTO

Este subsistema se encarga de introducir nuevos datos en el modelo de usuarios, a partir de una tabla de la base de datos seleccionada, de manera que enriquece el modelo de usuarios. Tiene un único cometido: el **almacenamiento de información**, obtiene datos de nuevos usuarios de una tabla de la base de datos elegida, los trata y los introduce en el modelo de usuarios seleccionado.

5) EVALUACIÓN

Subsistema encargado de estimar las certezas de una característica de cada uno de los usuarios seleccionado, habiendo aprendido desde 0 hasta el número de características que posea el usuario seleccionado en cada momento. Tiene un único objetivo, la **evaluación de información**, que mostrará por pantalla los datos de las certezas obtenidos en la estimación.

Una vez definidos los subsistemas de análisis se van a enumerar los casos de uso del sistema Entorno de trabajo con modelos de usuario. Para un mayor detalle de los mismos se puede acudir al ANEXO B. Aplicación Entorno de trabajo con modelos de usuario.

ID	<i>Nombre</i>
CU-001	Introducir nuevos datos
CU -002	Eliminar datos
CU -003	Adquirir nuevos datos
CU -004	Inferir un átomo de información
CU -005	Inferir varios átomos de información
CU -006	Comprobar un átomo de información
CU -007	Aprender inferencia
CU -008	Cargar fichero de configuración
CU -009	Almacenar datos de configuración
CU -010	Introducir datos de configuración
CU -011	Entrenar el modelo
CU -012	Evaluar usuarios con aprendizaje
CU -013	Cerrar sesión

Tabla 13. Casos de uso del Entorno de trabajo para modelos de usuario

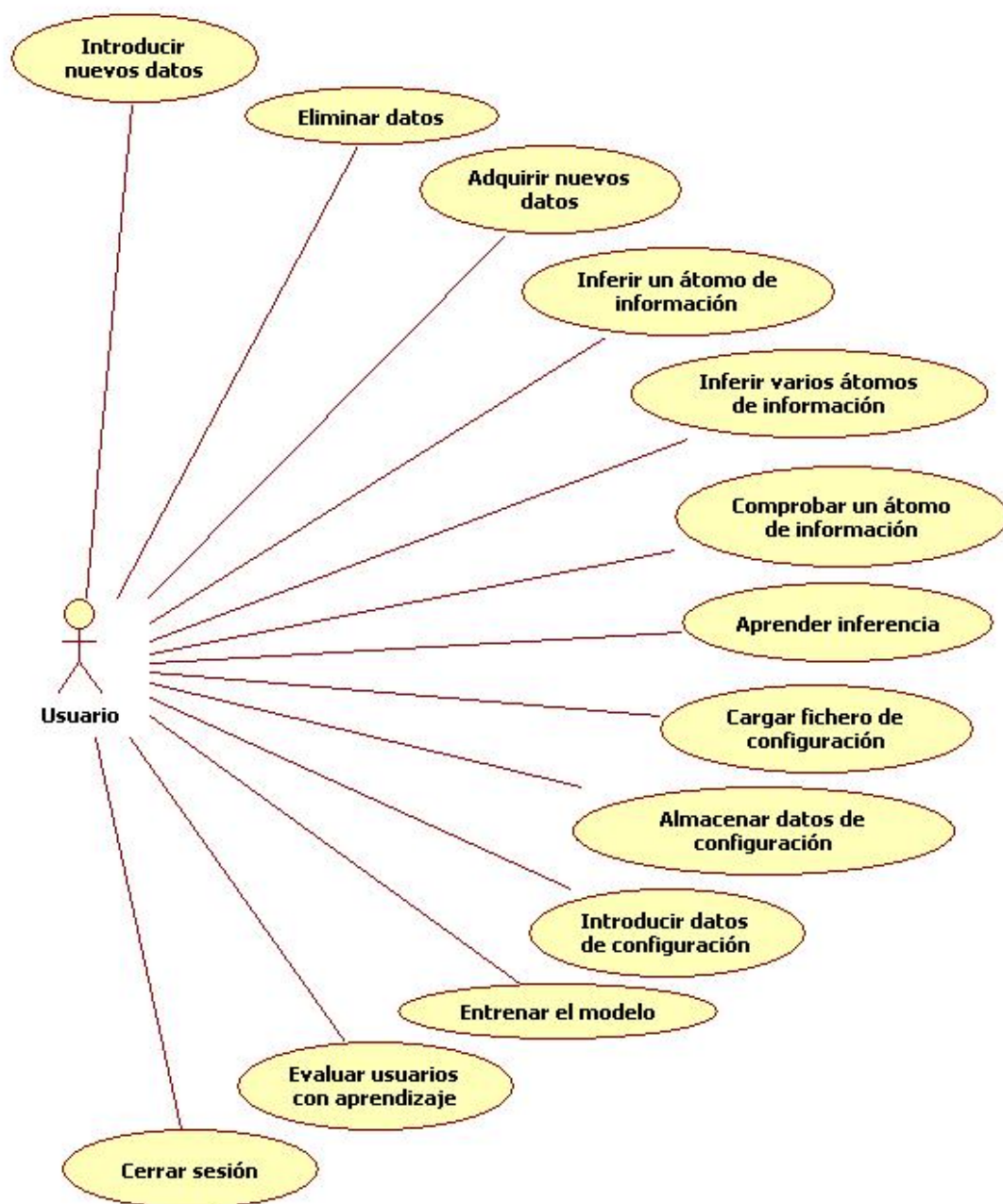


Ilustración 20. Diagrama de casos de uso del Entorno de trabajo para modelos de usuario

Seguidamente se presentan los diagramas de secuencia para cada uno de los casos de uso definidos, para ofrecer una visión de cómo será la sucesión de acciones para lograr el objetivo del caso de uso.

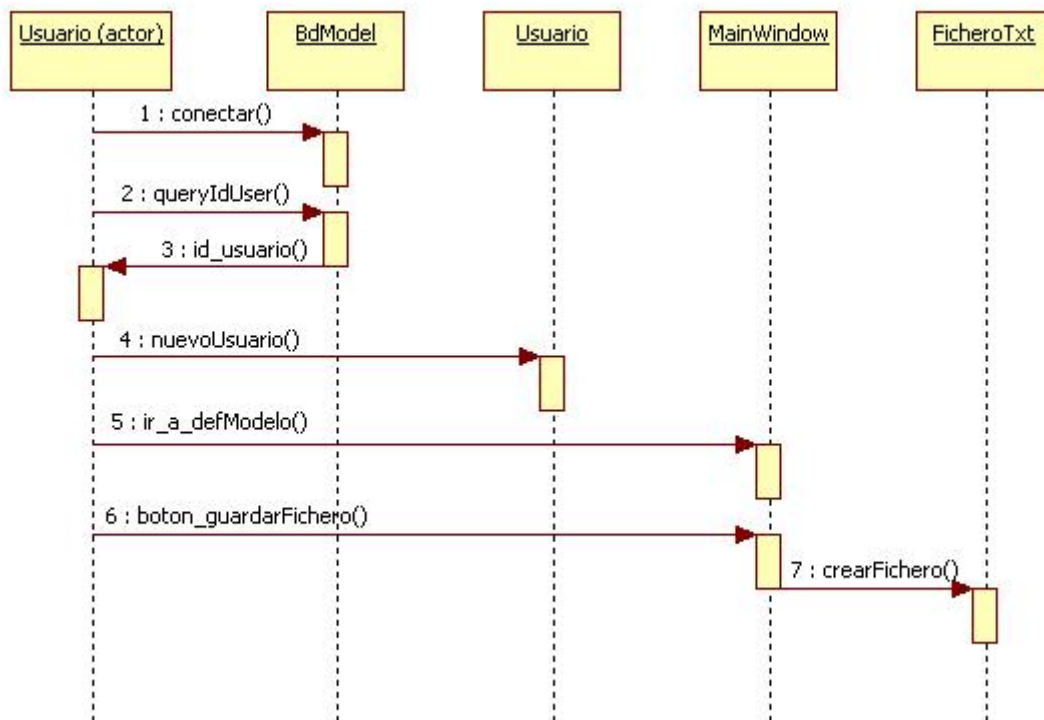


Ilustración 21. Diagrama de secuencia: Introducir nuevos datos

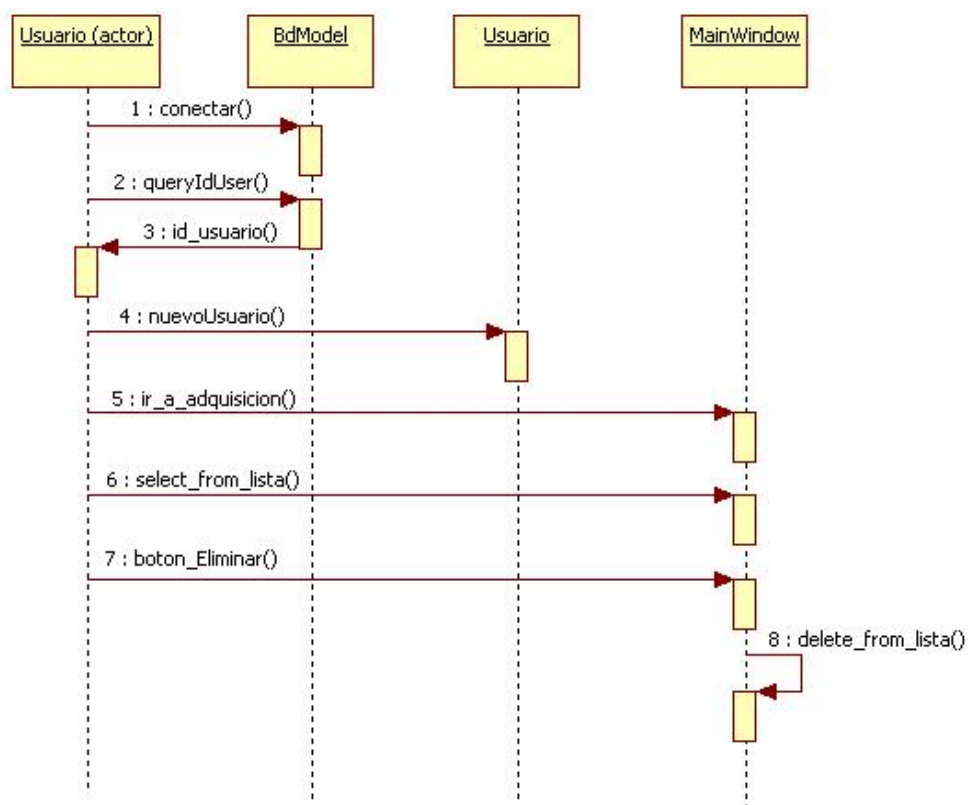


Ilustración 22. Diagrama de secuencia: Eliminar datos

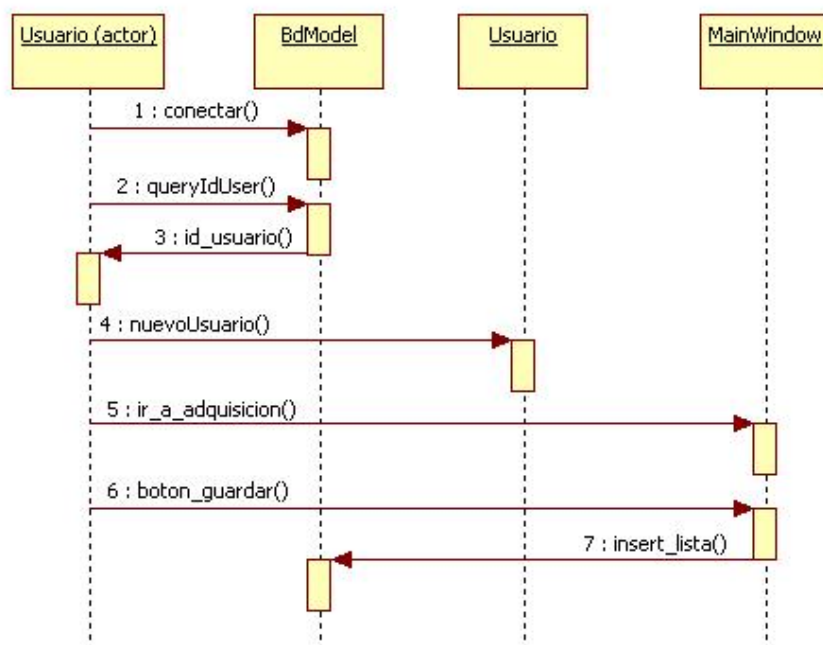


Ilustración 23. Diagrama de secuencia: Adquirir nuevos datos

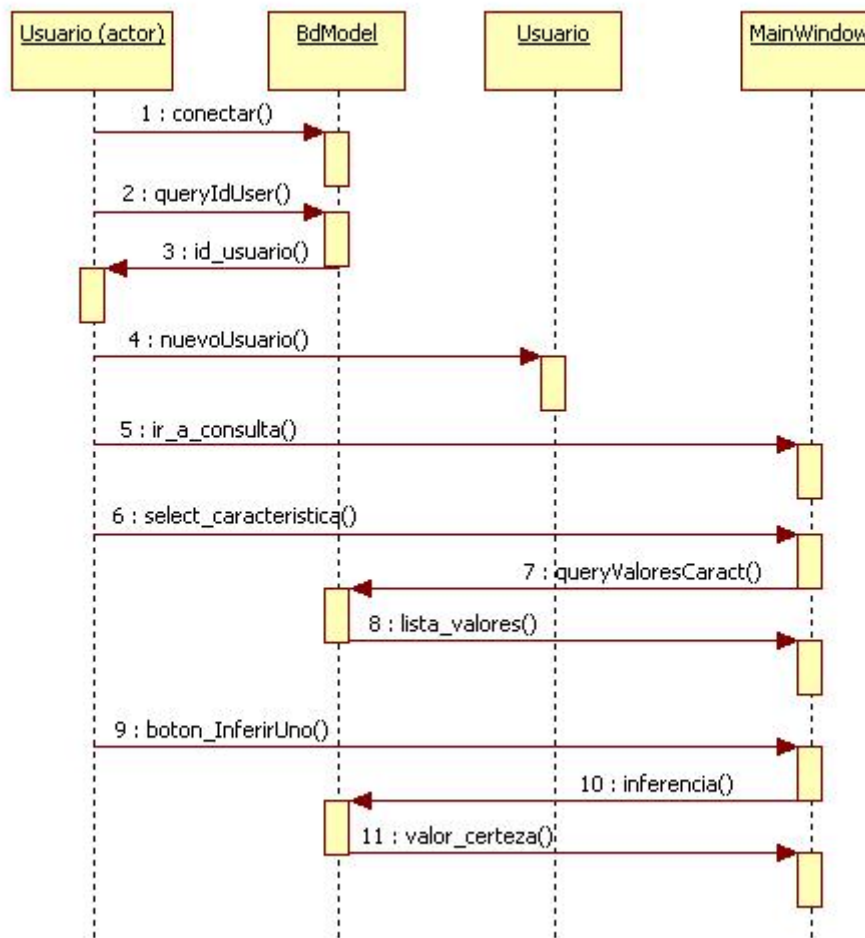


Ilustración 24. Diagrama de secuencia: Inferir un átomo de información

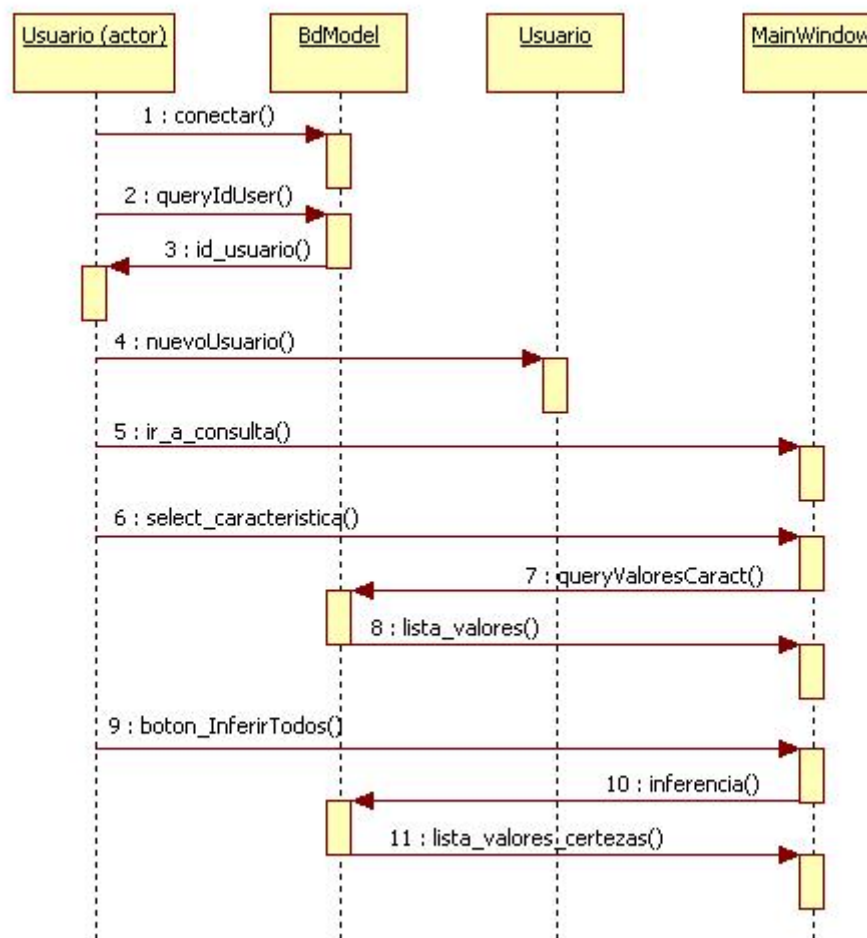


Ilustración 25. Diagrama de secuencia: Inferir varios átomos de información

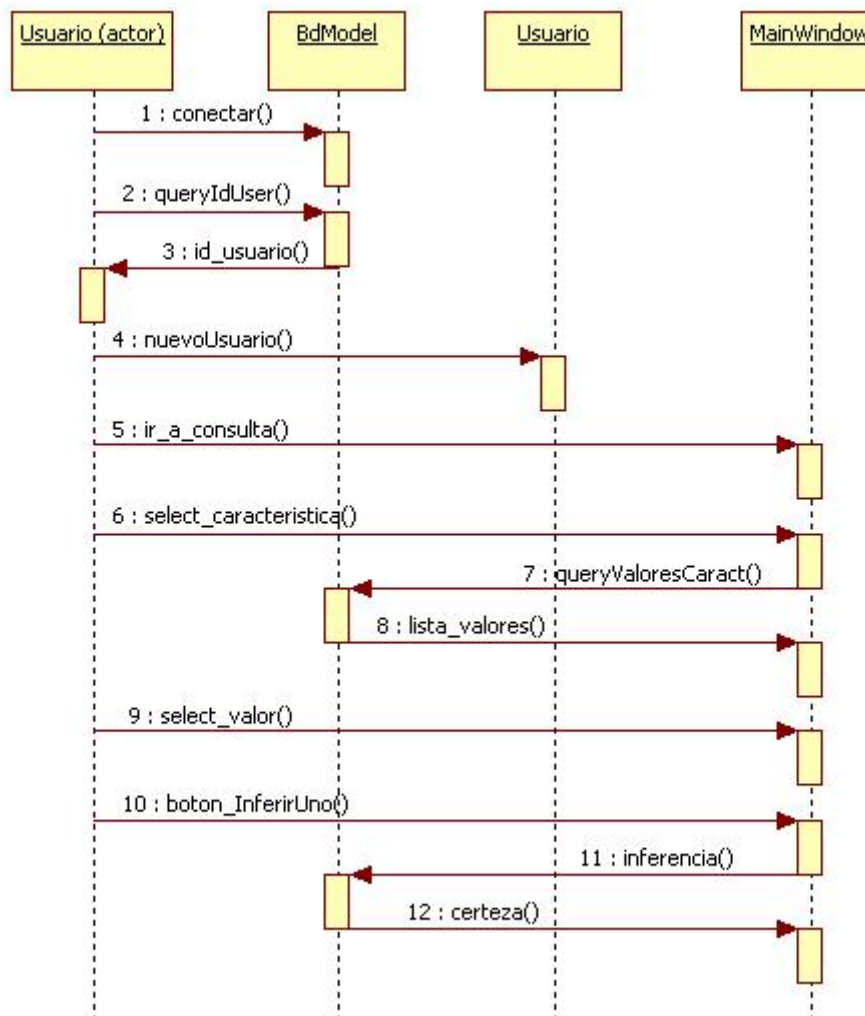


Ilustración 26. Diagrama de secuencia: Comprobar un átomo de información

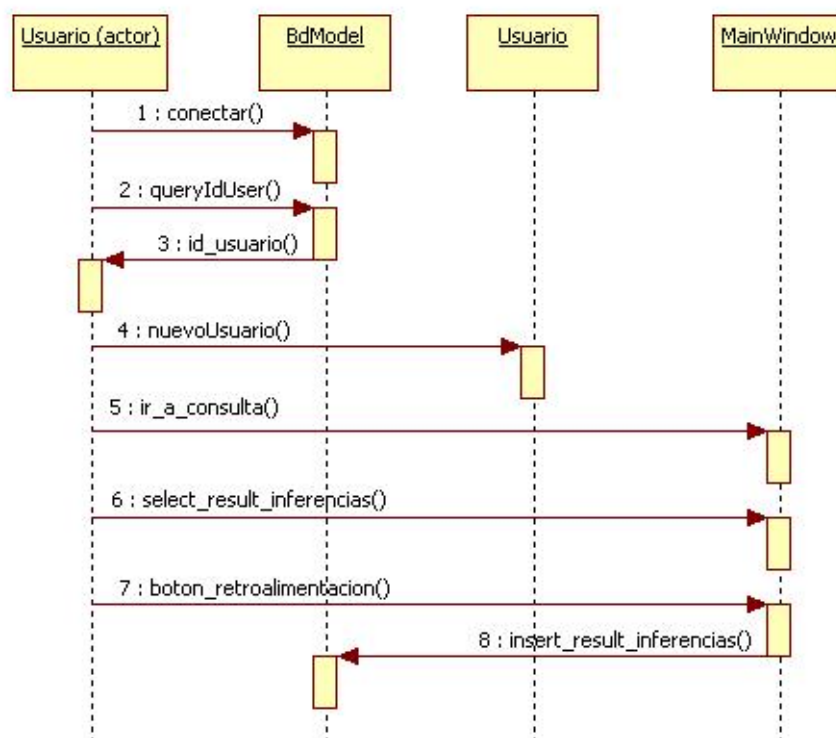


Ilustración 27. Diagrama de secuencia: Aprender inferencia

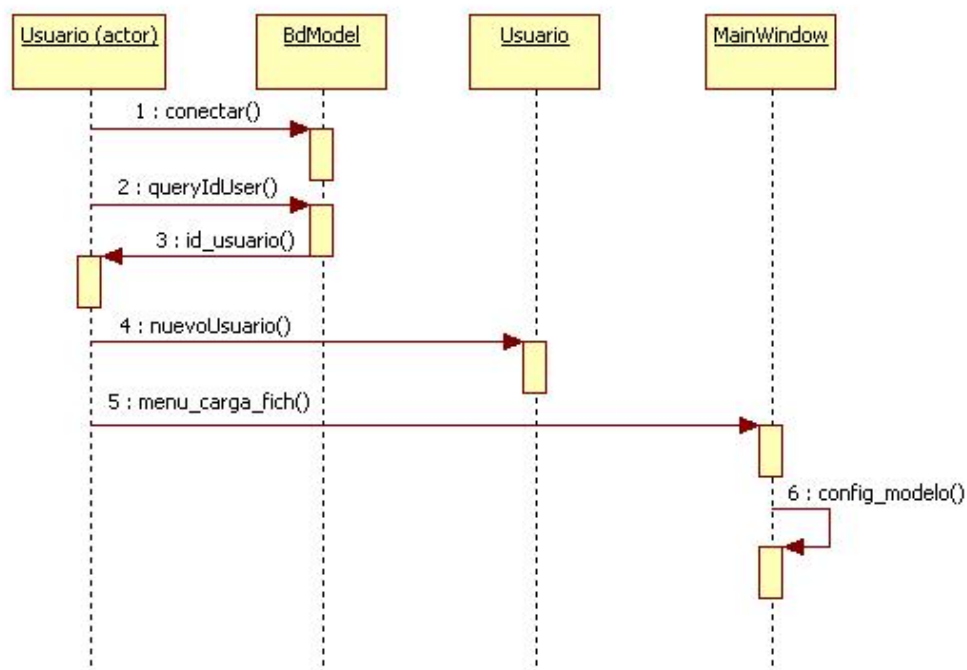


Ilustración 28. Diagrama de secuencia: Cargar fichero de configuración

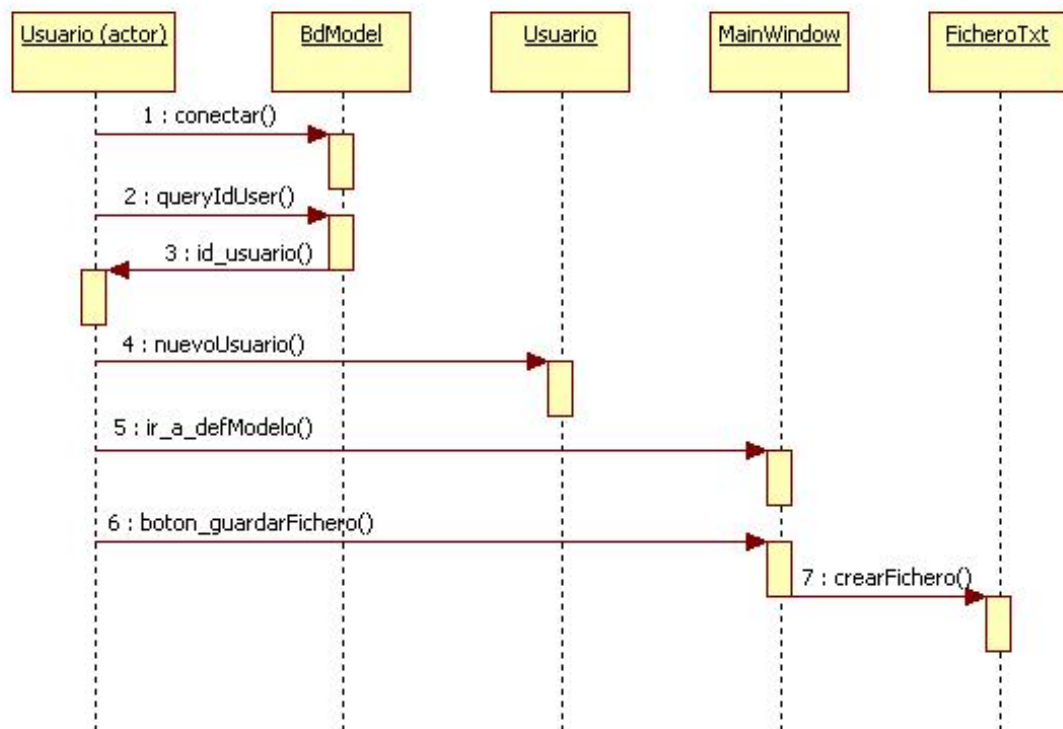


Ilustración 29. Diagrama de secuencia: Almacenar datos de configuración

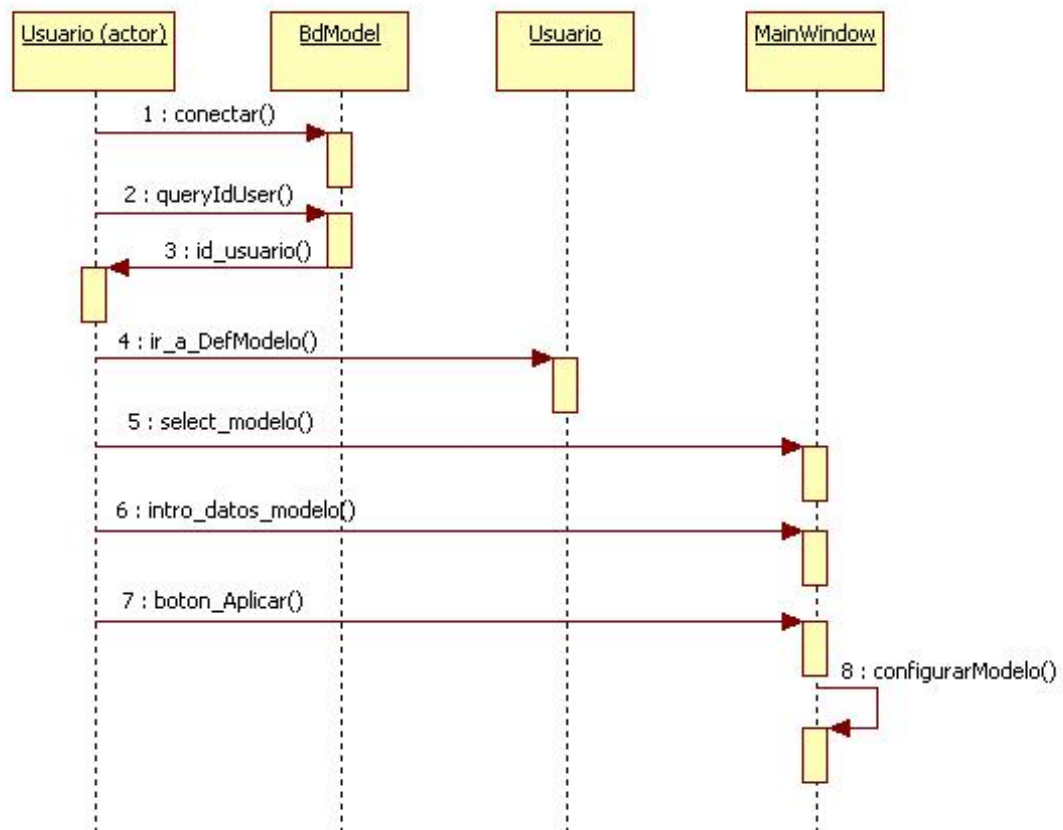


Ilustración 30. Diagrama de secuencia: Introducir datos de configuración

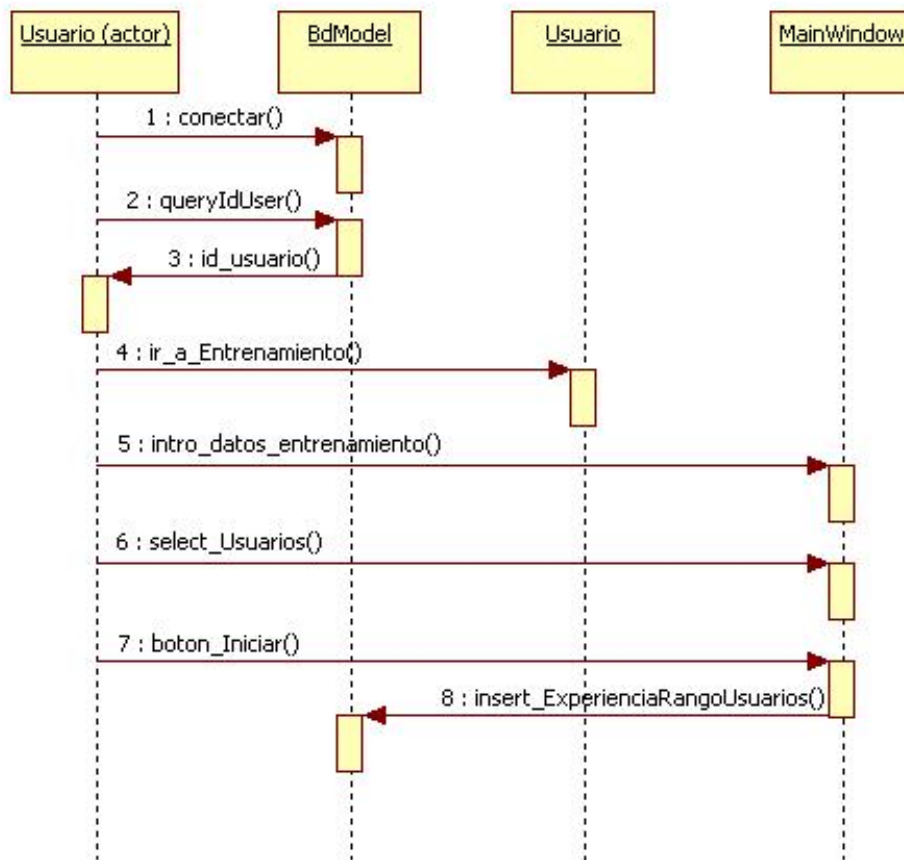


Ilustración 31. Diagrama de secuencia: Entrenar el modelo

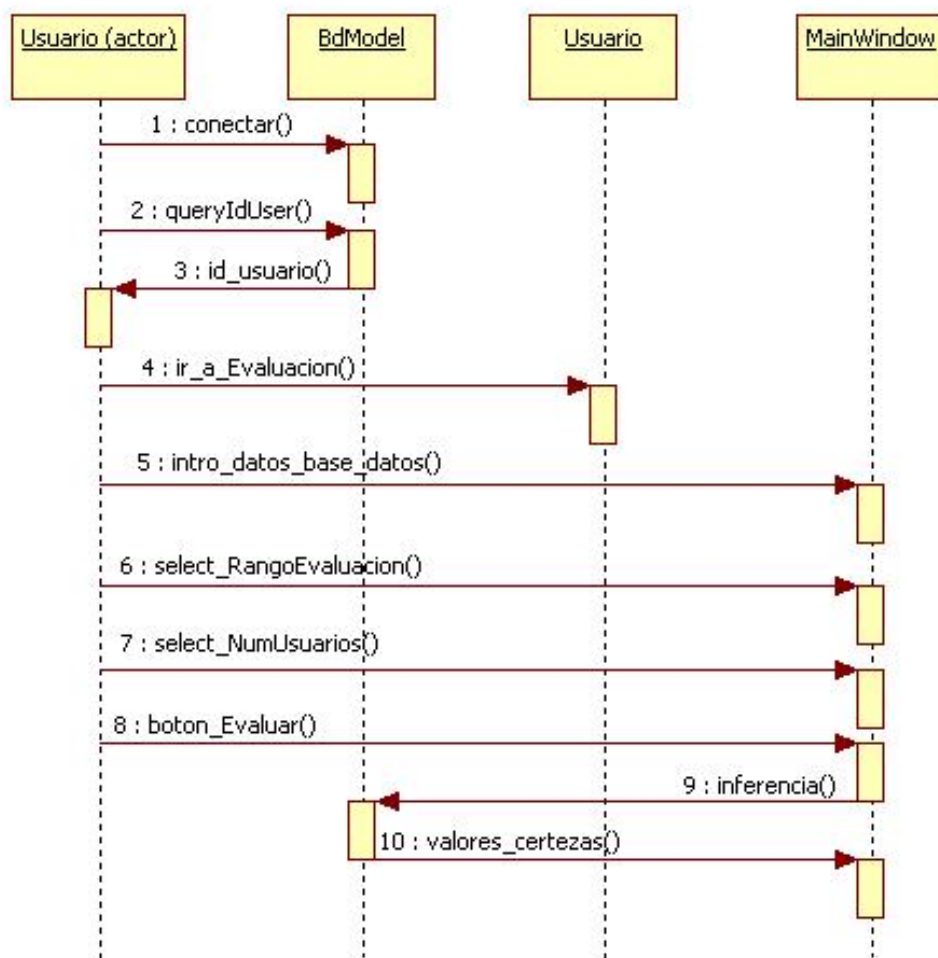


Ilustración 32. Diagrama de secuencia: Evaluar usuarios con aprendizaje

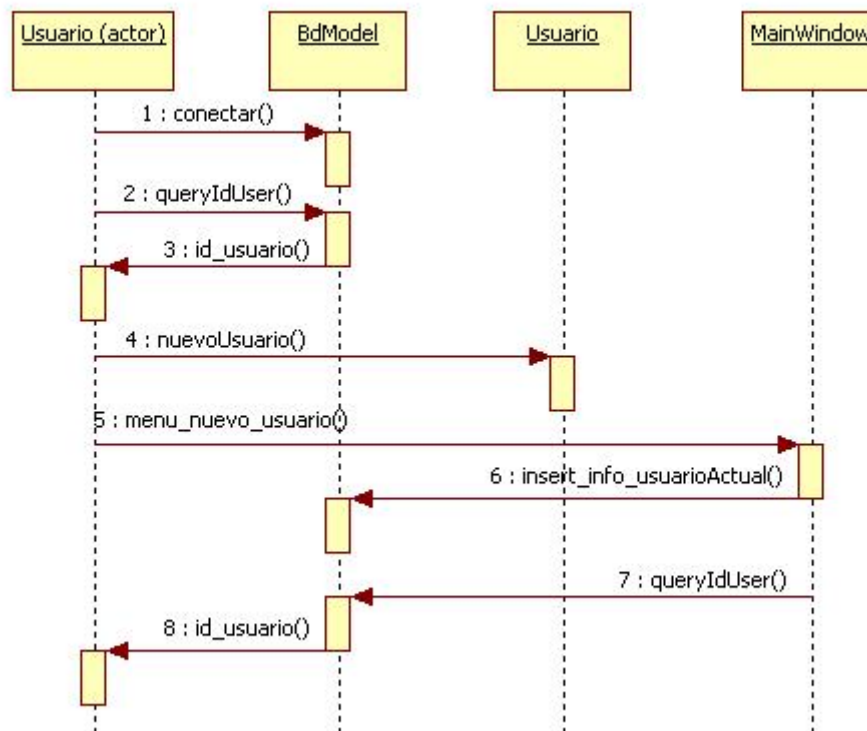


Ilustración 33. Diagrama de secuencia: Cerrar sesión

4.2.2. Necesidades de información

En este apartado se va a mostrar toda la información relevante al modelo conceptual de datos, descrito mediante un diagrama entidad-interrelación. A través de un estudio detallado se explicarán todas las interrelaciones posibles que se pueden llegar a dar entre cada una de las entidades existentes, las cuales también serán razonadas e ilustradas.

Como se viene comentando, no se trata de encontrar la mejor solución en general, ni de encontrar una buena solución a los problemas propuestos, sino de encontrar la mejor solución para este caso determinado. Dicha solución será obtenida gracias al completo conocimiento de los requerimientos y problemas planteados.

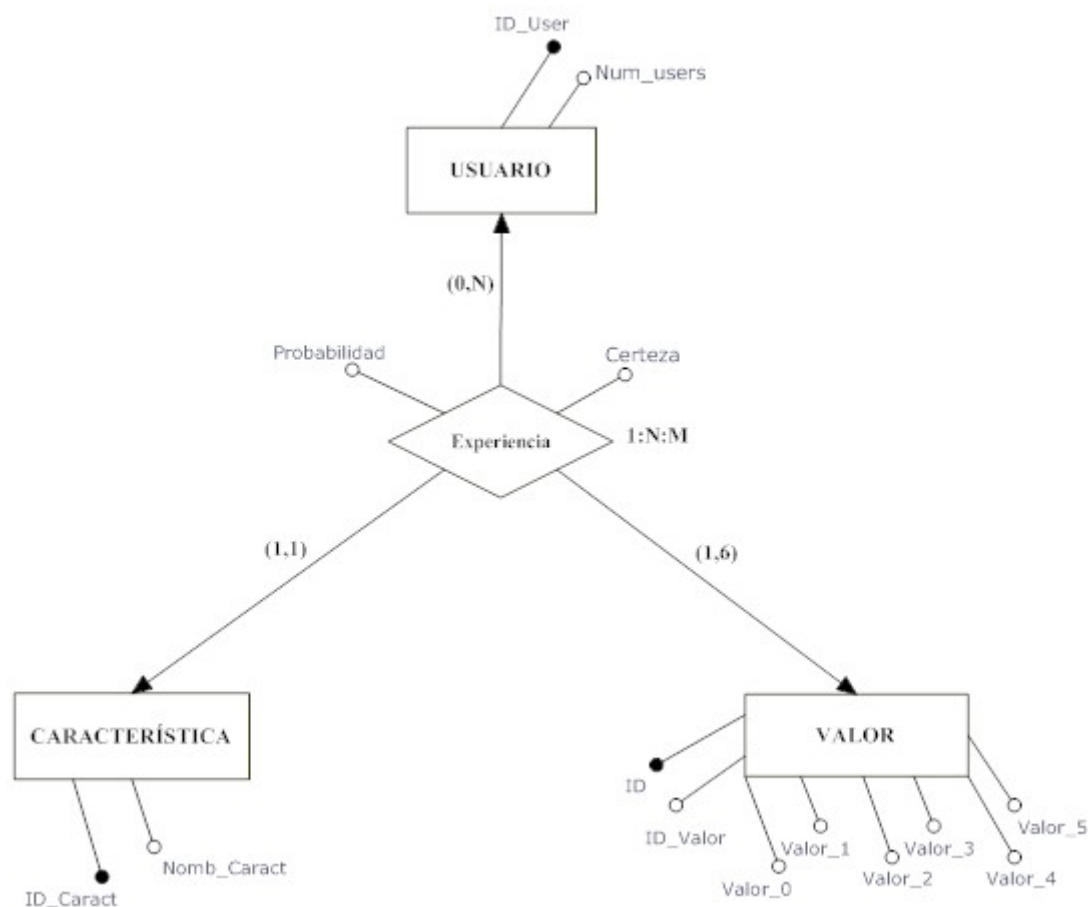


Ilustración 34. Modelo E/R Extendido

Como se puede observar en la figura anterior la base de datos de la aplicación consta de tres tablas: usuario, categoría y página., necesarias para almacenar la información que maneja la aplicación para cumplir sus objetivos.

Con esto se consigue un modelo consistente en una base de datos, que se actualizará fácilmente, debido a que cuenta con un número de tablas reducido, cuyos campos también han sido reducidos al mínimo número, eliminando posibles campos no necesarios.

1) ENTIDADES

En este apartado se explicará con todo detalle las tres entidades que componen el esquema E/R comentado con anterioridad (Ilustración 18):

- **USUARIO:** guarda los datos correspondientes a los grupos de usuarios existentes en el sistema, así como los datos del usuario actual que se hayan podido introducir en una sesión. Además del identificador correspondiente al grupo de usuarios o al usuario

actual se almacenará el número de usuarios que componen el grupo de usuarios, que puede ir desde 1 hasta n usuarios.

- **CARACTERÍSTICA:** contiene una lista de todas las características que puede tener un usuario (o grupo de ellos). Se puede referir a cualquier tipo de característica sobre el usuario.
- **VALOR:** encierra todos los posibles valores para las características definidas en la tabla “CARACTERÍSTICA”.

2) INTERRELACIONES

Como se puede observar a la vista del modelo entidad-relación extendido (figura 20) la base de datos está compuesta por una única interrelación: ternaria (‘Experiencia’, que relaciona las tres tablas existentes), detallada a continuación:

En el modelo definido se trata de una interrelación 1:N:M, pero, en general, esta interrelación ternaria debería ser N:M:P, ya que en dominios más amplios un valor podría pertenecer a más de una característica, como por ejemplo el valor ‘Castaño’ podría referirse al color de pelo o bien al color de ojos.

Al tratarse de una relación ternaria se estudiarán por separado cada una de las entidades participantes desde el punto de vista de la relación con el resto de entidades integrantes.

Para interpretar cómo interviene cada entidad en la relación se utiliza la técnica del 2–1, que consiste en dejar fijas dos entidades mientras se efectúan todas las posibles combinaciones con la entidad restante. A continuación se describen las tres posibilidades existentes:

- **USUARIO – CARATERÍSTICA / VALOR:** Dejando fijas las entidades de “USUARIO” y “CARACTERÍSTICA”, se probarán todas las combinaciones posibles de “VALOR”.

Un usuario determinado con una característica se podrá relacionar, como mínimo, con un registro de la entidad “valor”; ya que si un grupo tiene asignada una característica determinada tendrá que estar asociado al menos a un valor, porque sino no existiría esa relación de “experiencia”. Un usuario puede llegar a tener todas las características definidas en el sistema, pero puede no tener almacenados valores sobre todas ellas, y en la relación sólo se almacenarán las características con su valor determinado asociadas a un usuario.

- **USUARIO – VALOR / CARACTERÍSTICA:** Dejando fijas las entidades de “USUARIO” y “VALOR”, se probarán todas las combinaciones posibles de “CARACTERÍSTICA”.

Para un determinado usuario con un único valor, puede haber una y sólo una característica, puesto que en el presente dominio un valor determinado sólo puede estar referido a una característica.

- **CARACTERÍSTICA – VALOR / USUARIO:** Dejando fijas las entidades de “CARACTERÍSTICA” y “VALOR”, se probarán todas las combinaciones posibles de “USUARIO”.

Para una característica determinada y un valor para la misma, puede no referirse a grupo alguno, ya que la característica con sus valores correspondientes puede estar dada de alta en el sistema, pero que aún no haya usuarios que posean dicha característica y algún valor de la misma. Por otro lado, como máximo puede estar relacionado con un máximo de N usuarios, pudiendo existir tuplas de característica – valor iguales para distintos valores de usuarios.

3) SEMÁNTICA NO CONTEMPLADA

En el modelo entidad / relación, presentado en la Ilustración 18, no se recoge la siguiente semántica:

- Las características y los valores presentan una relación entre ellas, al margen de la relación “Experiencia”, y es que una característica puede contener de 1 hasta N valores. No se representa en el modelo, ya que no influye en el desarrollo del proyecto.
- En el dominio del sistema, un usuario asociado a una característica no va a presentar más de 5 valores por característica, ya que ésta es la cota máxima de información presentada en el perfil del usuario. Pero se deja abierto para posibles ampliaciones.
- Un usuario sólo tendrá asociadas 4 características como máximo, ya que, una vez más, en el perfil de usuario no aparece más información.

4.2.3. Interfaz

Se proporciona una ventana, construida en Java, que contará con las siguientes subinterfaces:

- **Adquisición:** contiene listas desplegables para poder seleccionar las características y sus posibles valores, además de 3 botones: Insertar, Eliminar y Guardar. En la parte inferior de la ventana aparecerá una lista, que se irá rellenando con pares de características-valor a medida que el usuario los vaya insertando en la misma.

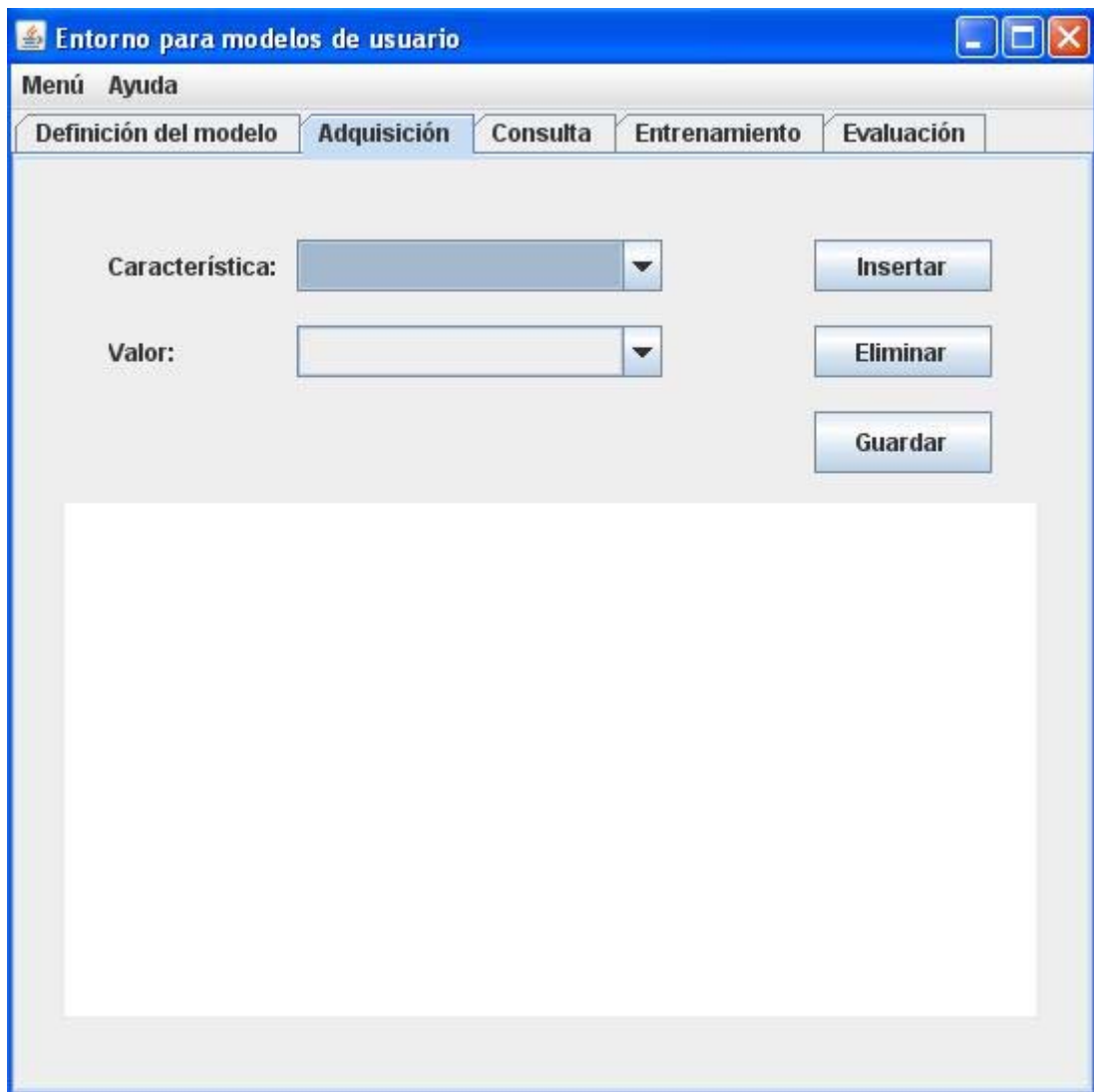


Ilustración 35. Interfaz Adquisición del Entorno de trabajo con modelos de usuario

- **Consulta:** contiene listas desplegables para poder seleccionar las características y sus posibles valores; también contiene 3 botones: Inferir uno, Inferir todos y Certeza. En la parte inferior de la ventana aparecerá una lista, que se rellenará con los datos inferidos. Además contiene otros dos botones en la parte más baja, “Retroalimentación positiva” y “Retroalimentación negativa”.



Ilustración 36. Interfaz Consulta del Entorno de trabajo con modelos de usuario

- **Entrenamiento:** posee varias cajas de texto, que permiten la recogida de distintos valores necesarios para el correcto funcionamiento de la aplicación, así como dos botones de radio que permiten seleccionar el rango de identificadores de usuario. Por último, contiene un botón, “Iniciar”.

Entorno para modelos de usuario

Menú Ayuda

Definición del modelo Adquisición Consulta Entrenamiento Evaluación

Nombre host: Usuario:

Base de datos origen: Contraseña:

Nombre de la tabla:

Selección de identificadores de usuarios:

☐ Todos

☐ Del al

Iniciar

Ilustración 37. Interfaz Entrenamiento del modelo del Entorno de trabajo con modelos de usuario

- **Definición del modelo:** incluye dos botones de radio para seleccionar el tipo de modelo, además de varias cajas de texto para introducir los datos del modelo seleccionado. En el extremo inferior de la ventana aparecen 3 botones: “Guardar en fichero”, “Aplicar” y “Reiniciar”.

Entorno para modelos de usuario

Menú Ayuda

Definición del modelo Adquisición Consulta Entrenamiento Evaluación

Selección del modelo:

☐ Oracle ☐ Java

Nombre host: Versión runtime: JRE 1.6.0 16
Usuario: Clase principal:
Contraseña:
Nombre paquete:
Nombre tabla:

Renombrado de métodos:

1. Adquirir un átomo de información	<input type="text"/>
2. Inferir un átomo de información	<input type="text"/>
3. Inferir varios átomos de información	<input type="text"/>
4. Comprobar un átomo de información	<input type="text"/>
5. Cerrar sesión (aprendiendo)	<input type="text"/>

Ilustración 38. Interfaz Definición del modelo del Entorno de trabajo con modelos de usuario

- **Evaluación:** encierra dos botones de radio para seleccionar el rango de evaluación y de 3 cajas de texto para que el usuario introducir los datos solicitados. También aparece una caja con flechas para establecer un valor numérico (número de usuarios a evaluar), seguido de un botón, “Evaluar”. Para finalizar se tiene un área de texto donde se mostrarán los resultados de la evaluación.

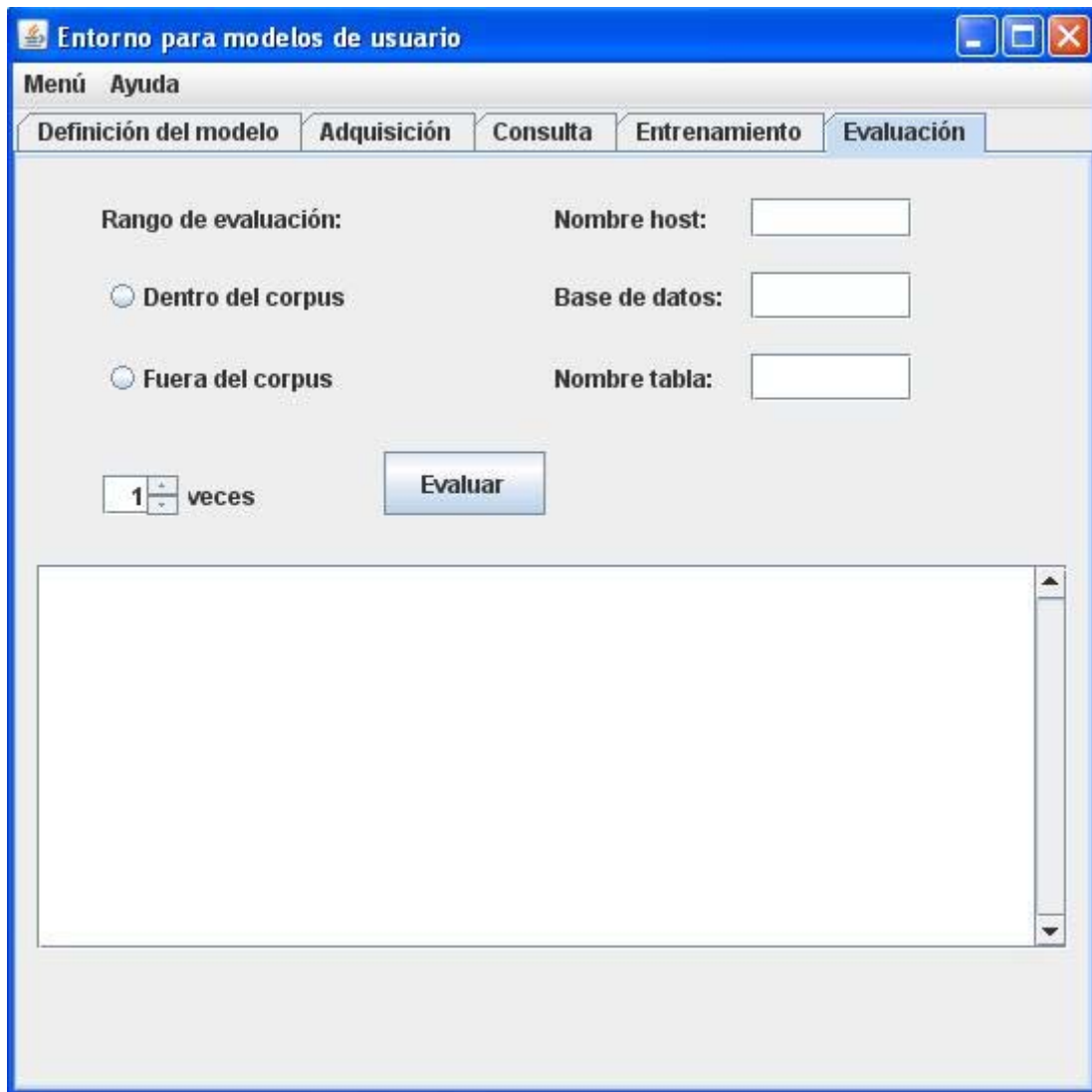


Ilustración 39. Interfaz Evaluación del Entorno de trabajo con modelos de usuario

- **Barra de menú:** que contiene dos opciones “Menú” y “Ayuda”. Dentro de la opción “Menú” aparecen dos sub-opciones: “Cerrar sesión” y “Cargar fichero de configuración”.

IMPLEMENTACIÓN

Codificando...

5. IMPLEMENTACIÓN

Una vez concluido todo el proceso de diseño y que han quedado definidos todos los algoritmos necesarios para el desarrollo de la aplicación, ahora nos disponemos a codificar toda esa información tratada en las fases anteriores, creando un sistema en consecuencia. En esta etapa se tienen que traducir dichos algoritmos a un lenguaje de programación específico, es decir, las acciones definidas en los algoritmos hay que convertirlas a instrucciones.

Para llevar a cabo una codificación exitosa es necesario seguir un orden preestablecido, comprobando constantemente que lo que se está codificando se ajusta por completo al diseño realizado con anterioridad.

Aunque puede parecer que al tener hecho el diseño del sistema la fase de codificación sea un mero trámite no es así, para codificar un algoritmo hay que conocer la sintaxis del lenguaje que se va a traducir. Sin embargo, independientemente del lenguaje de programación en que esté escrito un programa, será su algoritmo el que determine su lógica; dicha lógica establece cuáles son las acciones del programa y en qué orden se deben ejecutar.

Para que el lector pueda comprender de manera clara el desarrollo de esta codificación se mostrarán todos los pasos seguidos para llevar a cabo la transformación del modelo entidad – relación al modelo relacional correspondiente. La obtención del modelo relacional nos permitirá codificarlo en lenguaje SQL, especificando todas las posibilidades y opciones analizadas.

Para finalizar el capítulo se explicarán todos los procesos, sin los cuales no sería posible la correcta ejecución del modelo. No se mostrará el código fuente de todos los procesos que intervienen, puesto que resulta demasiado extenso para incluirlo en este documento. En cualquier caso, queda a disposición del lector solicitarlo a la autora.

5.1. WEB CRAWLER

5.1.1. Diagrama de clases

El siguiente modelo explica los conceptos más significativos del dominio del problema. Después se comentarán algunos detalles de dicho modelo.

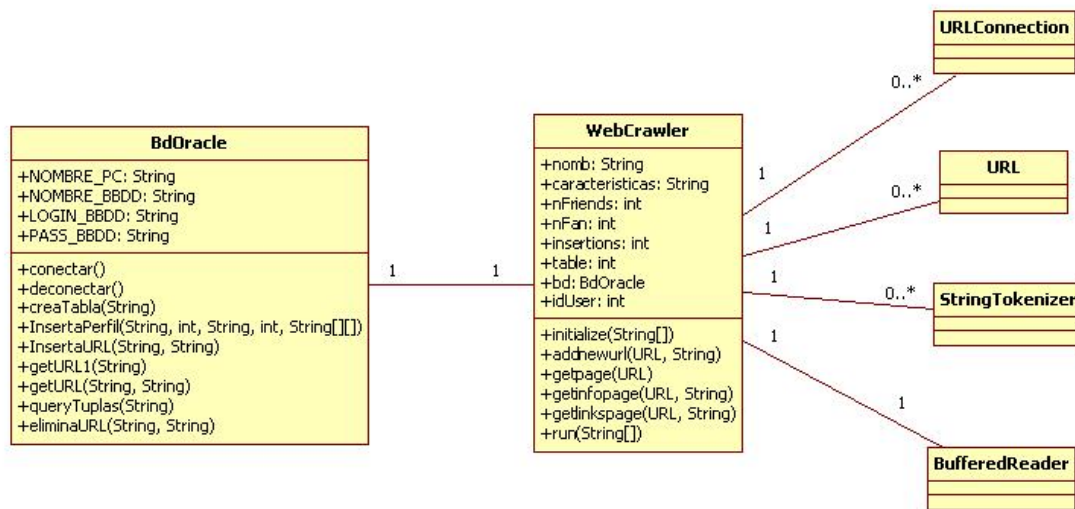


Ilustración 40. Diagrama de clases del Web Crawler

Se comentan a continuación las características de cada una de las clases especificadas, así como la necesidad de la existencia de cada una de ellas:

- **BdOracle**: identifica a la base de datos del sistema, donde se almacenan las direcciones Web (visitadas y pendientes de visitar) y la información extraída de las páginas Web procesadas por el robot de búsqueda.
- **WebCrawler**: representa el robot de búsqueda que va escaneando páginas Web del contexto y obteniendo la información que necesita para el cumplimiento de los requisitos del sistema.
- **URLConnection**: clase de Java necesaria para poder acceder al contenido de una página Web.
- **URL**: clase de Java que permite representar direcciones Web, de manera que la Web a la que enlazan pueda ser accesible desde el mismo programa Java que lo ejecute.
- **StringTokenizer**: clase de Java necesaria para poder recorrer el contenido descargado de una página Web y analizarlo de la manera en que se indica en los requisitos.
- **BufferedReader**: clase de Java que representa un buffer de lectura, necesario para realizar la descarga del contenido de la página Web.

5.1.2. Necesidades de información

Partiendo del modelo E/R Extendido definido anteriormente (Ilustración 18) se realizará la transformación al modelo relacional, para poder codificarlo posteriormente en lenguaje SQL (codificación de la base de datos).

Además, se especificarán una serie de técnicas necesarias para poder mantener íntegramente todas las opciones especificadas en el diseño del sistema.

Para comenzar se explican los distintos pasos seguidos para la transformación de los elementos básicos (entidades, atributos, interrelaciones...) necesaria para la realización del grafo relacional. Después se muestra el grafo relacional propiamente dicho con los consiguientes comentarios explicativos al mismo.

- **Entidades**: todo tipo de entidad del modelo E/R se convierte en una relación del modelo Relacional.

- **Atributos**: los atributos de una entidad serán atributos de la relación correspondiente, con ciertas salvedades que se detallan a continuación: los atributos identificadores principales serán marcados como clave primaria, mientras que los atributos identificadores alternativos serán marcados como claves alternativas.

- **Interrelaciones**: todo tipo de **interrelación 1:N** da lugar al fenómeno de propagación de clave (clave ajena: aunque parece recoger menos semántica que una relación E/R, esta semántica se complementa con la que aporta la restricción referencial). Se propaga el atributo identificador principal del tipo de entidad que tiene de cardinalidad máxima 1 a la que tiene de cardinalidad máxima N. Otra opción posible para este tipo de interrelación es transformarla 1:N en una nueva relación, que tendría como atributos las claves de ambas entidades (y los atributos propios de la interrelación), siendo su clave la misma que la que posee la entidad que interviene en la interrelación con N ocurrencias.

El tipo de **interrelación ternaria** (relación en la que participan 3 entidades) se representará a través de relaciones binarias siempre que sea posible; en caso contrario, como regla general, se representará mediante una relación intermedia, tal que dicha relación contendrá como clave primaria la concatenación de los atributos principales de las entidades participantes y si la interrelación correspondiente tiene algún atributo éste será incluido en la relación creada para la representación. Ahora bien, para el caso particular de que dos ramas tengan como cardinalidad máxima N y 1 en la rama restante, la relación intermedia que lo

representa tendrá como clave primaria la concatenación de los atributos principales de las dos entidades que participan en la relación con cardinalidad máxima N.

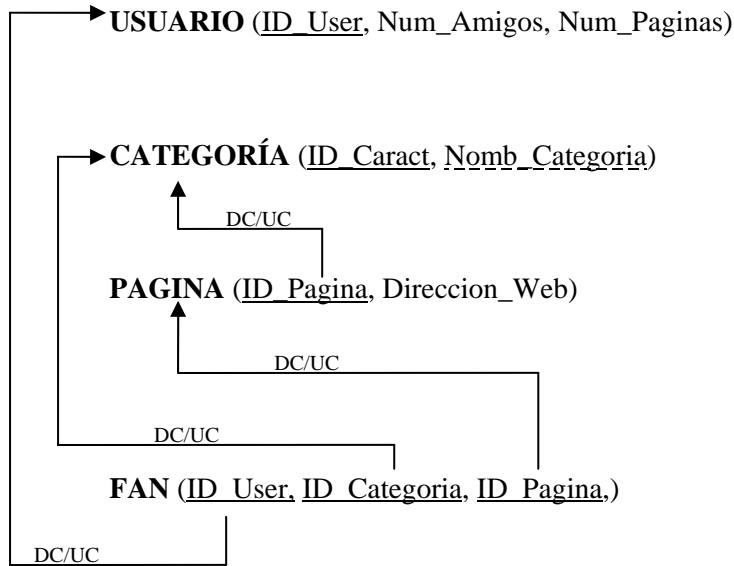


Ilustración 41. Gráfico relacional del Web Crawler

Para una mejor comprensión del grafo por parte de lector ahora se va a explicar algunas notaciones empleadas en dicho grafo:

- El único modo de borrado y actualización empleado es el DC/UC (*Delete Cascade / Update Cascade*), es decir, que a la hora de efectuar borrados y modificaciones en la tabla principal estas acciones también se llevarán a cabo en la tabla secundaria.

Existen otros modos de borrado y actualización que no ha sido necesario utilizar para el proyecto, pero que se comentarán a continuación:

- DR/UR (*Delete Restrict / Update Restrict*): modo restringido: no permite una eliminación si hay un registro asociado.
- DSD/USD (*Delete Set Default / Update Set Default*): con valor por defecto: al producirse la eliminación el registro asociado pasa a tener un valor por defecto.
- DSN/USN (*Delete Set Null / Uptade Set Null*): con valor nulo: al producirse la eliminación el registro asociado contendrá valor nulo.

- Las claves ajenas (*Foreign Keys*) no se han especificado en el diagrama para no complicar su legibilidad, por lo que serán especificadas a continuación:

- La relación PÁGINA cuenta con una clave ajena, que recibe de la entidad CATEGORÍA.
- La relación FAN posee tres claves ajenas, que provienen de las tres entidades que relaciona: USUARIO, CATEGORÍA y PÁGINA.

Una vez comprendida la notación empleada en el grafo se va a analizar y explicar cada una de las decisiones tomadas en la transformación.

Como se comenta en la introducción a esta transformación cada entidad del modelo E/R ha sido transformada en una relación del modelo relacional, sin complicación alguna. En la misma línea han sido transformados los atributos de cada entidad existente.

Ahora bien, en el caso de las interrelaciones ha sido necesario tomar decisiones importantes a la hora de transformar la interrelación que compone el modelo E/R: para representar la interrelación ha sido necesaria la creación de una nueva relación: FAN, debido a que no es posible representar dicha interrelación mediante relaciones binarias. Entonces la tabla FAN contiene las claves primarias de las relaciones que participan con cardinalidad máxima N, que son USUARIO, CATEGORÍA y PÁGINA. Dado que la clave primaria de esta nueva entidad no es más que la concatenación de las claves primarias de las relaciones comentadas, dicha clave primaria está compuesta por claves ajenas. Por ello, la actualización y el borrado de datos se realizarán en cascada.

Sin embargo, con la creación de la tabla FAN comentada no quedan reflejados todos los casos posibles, ya que puede haber páginas que se correspondan a categorías, sin que tengan que estar obligatoriamente relacionados con un usuario. Por ello será necesario añadir en el esquema E/R una interrelación binaria entre CATEGORÍA y PÁGINA, tal y como se puede observar en la siguiente figura (Ilustración 42):

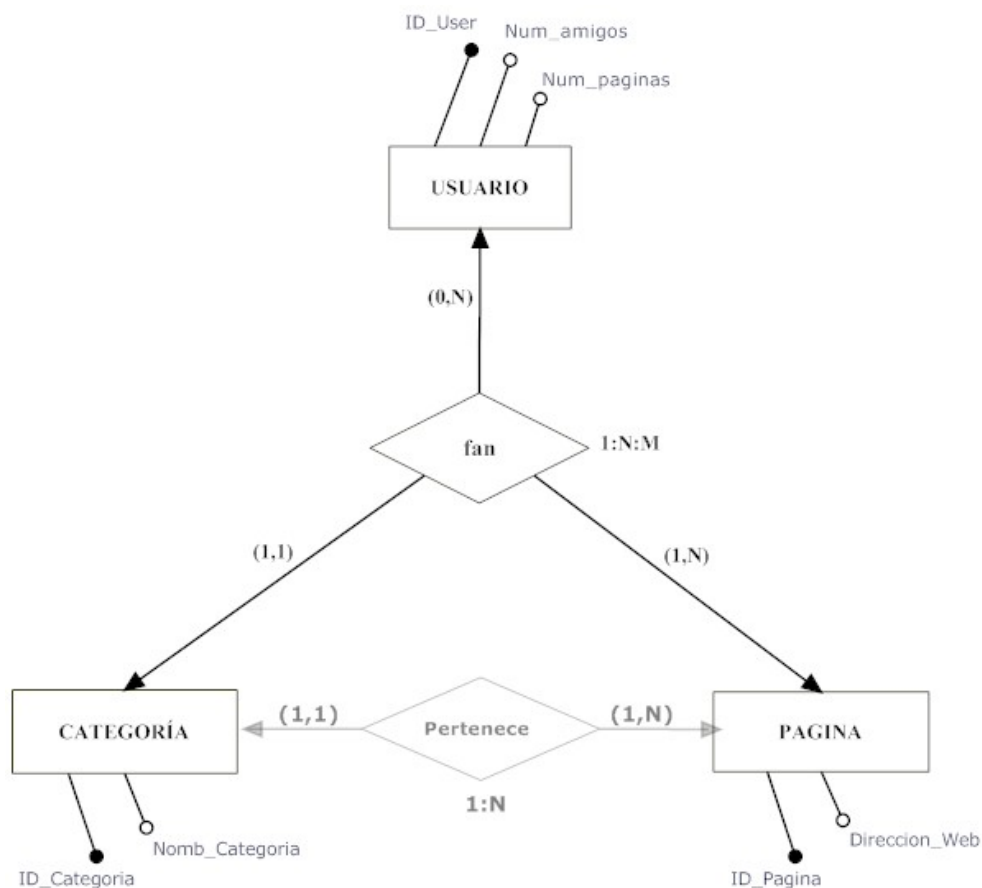


Ilustración 42. Modelo E/R Extendido del Web Crawler, preparado para la transformación al relacional

A continuación se presenta el grafo relacional que representa la nueva relación 'Pertenece' (Ilustración 43), fruto de la existencia de categorías y sus páginas correspondientes y que no están relacionados con usuario alguno:

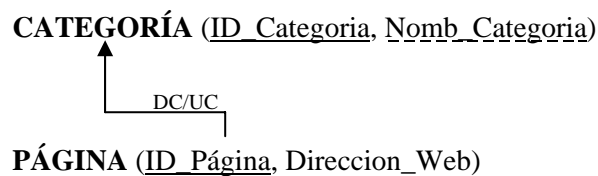


Ilustración 43. Grafo relacional de la interrelación 'Pertenece'

No es necesario crear una nueva tabla para representar la relación, porque, tal y como se comentó al inicio de este capítulo, se puede transformar propagando la clave, dando lugar a la creación de una clave ajena en la relación PÁGINA. El borrado y la actualización de esta clave ajena será en cascada, es decir, que la eliminación de una tupla de la relación CATEGORÍA implica no tener ninguna referencia suya en la relación PÁGINA y que la modificación de una tupla en esa relación también será modificada en la relación PÁGINA.

5.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO

5.2.1. Diagrama de clases

El modelo de la página siguiente explica los conceptos más significativos del dominio del problema. Después se comentarán algunos detalles de dicho modelo.

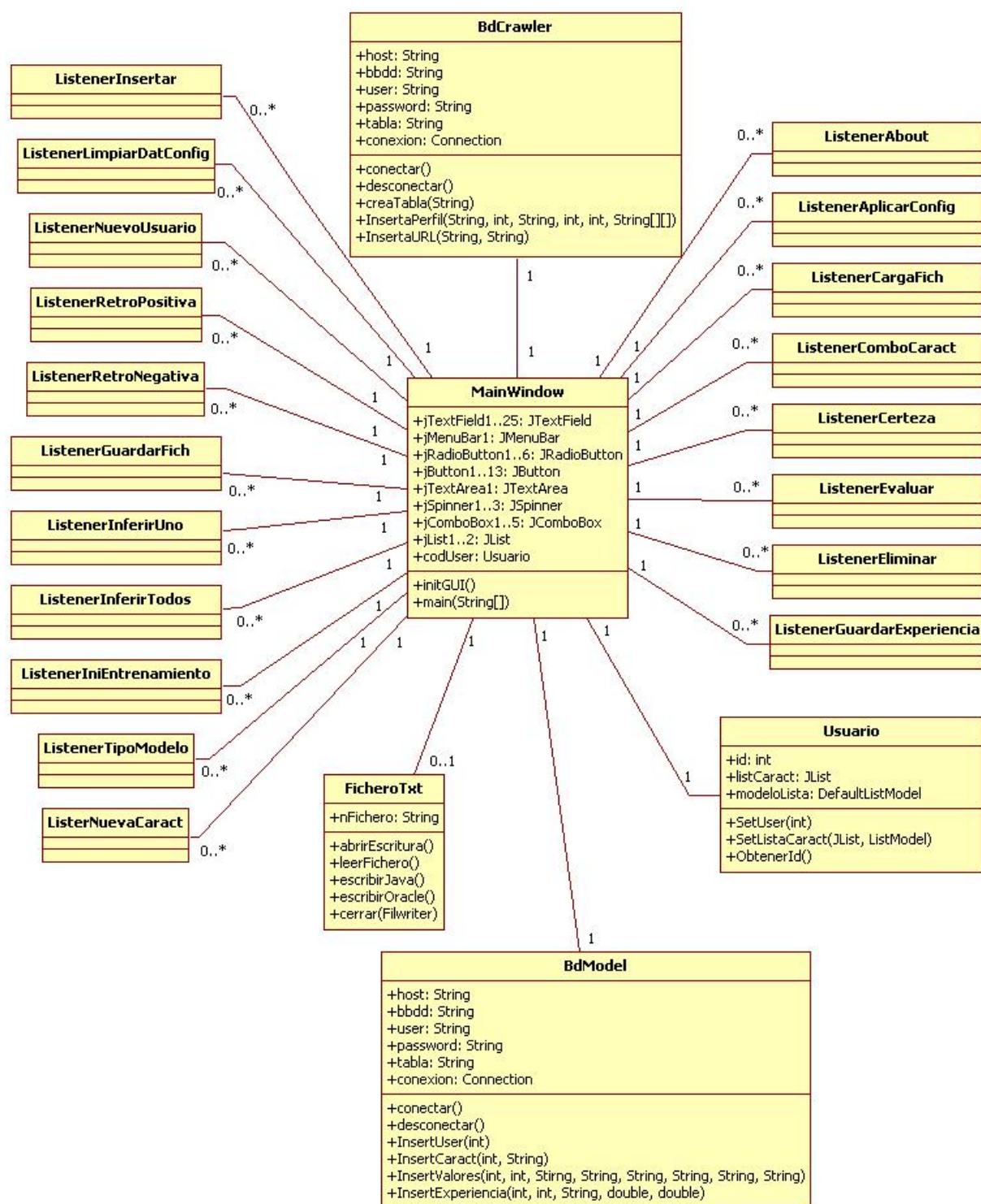


Ilustración 44. Diagrama de clases del Entorno de trabajo con modelos de usuario

Se comentan a continuación las características de cada una de las clases especificadas, así como la necesidad de la existencia de cada una de ellas:

- **BdCrawler**: identifica a la base de datos de donde se va a extraer información para entrenar el modelo de usuarios de la aplicación.
- **BdModel**: se refiere a la base de datos del modelo de usuario, donde se almacenan y se consultan todos los datos contenidos en el modelo de usuarios.
- **MainWindow**: representa la interfaz del sistema; implementa una ventana en Java y contiene todo lo especificado con anterioridad en el apartado de ENTORNO DE TRABAJO CON MODELOS DE USUARIO.
- **Usuario**: clase de Java que representa un usuario en la aplicación; queda definido por un identificador y su lista de características y valores.
- **FicheroTxt**: clase de Java que permite trabajar con ficheros de texto. Proporciona métodos para su lectura y escritura.
- **ListenerAbout**: clase que instancia un objeto de tipo ActionListener que se registra sobre la opción del menú “Acerca de”. Necesaria para poder realizar la acción deseada al acceder al menú “Acerca de”.
- **ListenerAplicarConfig**: crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Aplicar” de la pestaña “Definición del modelo”.
- **ListenerCargaFich**: clase que instancia un objeto de tipo ActionListener que se registra sobre la opción del menú “Cargar fichero de configuración”. Necesaria para poder realizar la acción deseada al acceder a la opción del menú comentado.
- **ListenerComboCaract**: crea una instancia de un objeto de tipo ActionListener que se registra sobre la lista desplegable de características, permitiendo ejecutar la acción necesaria cada vez que se modifique la selección de la lista desplegable.
- **ListenerCerteza**: clase que instancia un objeto de tipo ActionListener que se registra sobre el botón “Certeza” de la pestaña “Consulta”.
- **ListenerEvaluar**: crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Evaluar” de la pestaña “Evaluación” y que permite la ejecución de los comandos requeridos al pulsar dicho botón.

- **ListenerEliminar:** clase que instancia un objeto de tipo ActionListener que se registra sobre el botón “Eliminar” de la pestaña “Adquisición”. Implementa los métodos necesarios para que el botón realice su funcionalidad.
- **ListenerGuardarExperiencia:** crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Guardar” de la pestaña “Adquisición” y que permite la ejecución de los comandos requeridos al pulsar dicho botón.
- **ListenerInsertar:** clase que instancia un objeto de tipo ActionListener que se registra sobre el botón “Insertar” de la pestaña “Adquisición” y que permite la ejecución de los comandos requeridos al pulsar dicho botón.
- **ListenerLimpiarDatConfig:** crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Reiniciar” de la pestaña “Definición del modelo” y que implementa los métodos necesarios para que el botón realice su funcionalidad.
- **ListenerNuevoUsuario:** clase que instancia un objeto de tipo ActionListener que se registra sobre la opción del menú “Cerrar sesión” y que permite la ejecución de los comandos requeridos al pulsar el botón comentado.
- **ListenerRetroPositiva:** crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Reiniciar” de la pestaña “Definición del modelo” y que implementa los métodos necesarios para que el botón realice su funcionalidad.
- **ListenerRetroNegativa:** clase que instancia un objeto de tipo ActionListener que se registra sobre el botón “Retroalimentación negativa” de la pestaña “Adquisición” y que permite la ejecución de los comandos requeridos al pulsar el botón comentado.
- **ListenerGuardarFich:** crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Retroalimentación positiva” de la pestaña “Adquisición” y que permite la ejecución de los comandos requeridos al pulsar el botón comentado.
- **ListenerInferirUno:** clase que instancia un objeto de tipo ActionListener que se registra sobre el botón “Inferir uno” de la pestaña “Consulta” y que permite la ejecución de los comandos requeridos al pulsar dicho botón.
- **ListenerInferirTodos:** crea una instancia de un objeto de tipo ActionListener que se registra sobre el botón “Inferir todos” de la pestaña “Consulta” y que implementa los métodos necesarios para que el botón realice su funcionalidad.

- **ListenerIniEntrenamiento:** clase que instancia un objeto de tipo ActionListener que se registra sobre el botón “Iniciar” de la pestaña “Entrenamiento” y que permite la ejecución de los comandos requeridos al pulsar el botón comentado.
- **ListenerTipoModelo:** crea una instancia de un objeto de tipo ItemListener que se registra sobre el botón “Reiniciar” de la pestaña “Definición del modelo” y que implementa los métodos necesarios para que el botón realice su funcionalidad.
- **ListenerNuevaCaract:** clase que instancia un objeto de tipo ItemListener que se registra sobre la lista desplegable de características de la “Adquisición”, siempre que ésta tome el valor “Nuevo”. Permite la ejecución de los comandos requeridos al seleccionar el valor comentado.

5.2.2. Necesidades de información

Partiendo del modelo E/R definido con anterioridad (Ilustración 34) se realizará la transformación al modelo relacional, para poder codificarlo posteriormente en lenguaje SQL (codificación de la base de datos).

Seguidamente se muestra el grafo relacional propiamente dicho con los consiguientes comentarios explicativos al mismo.

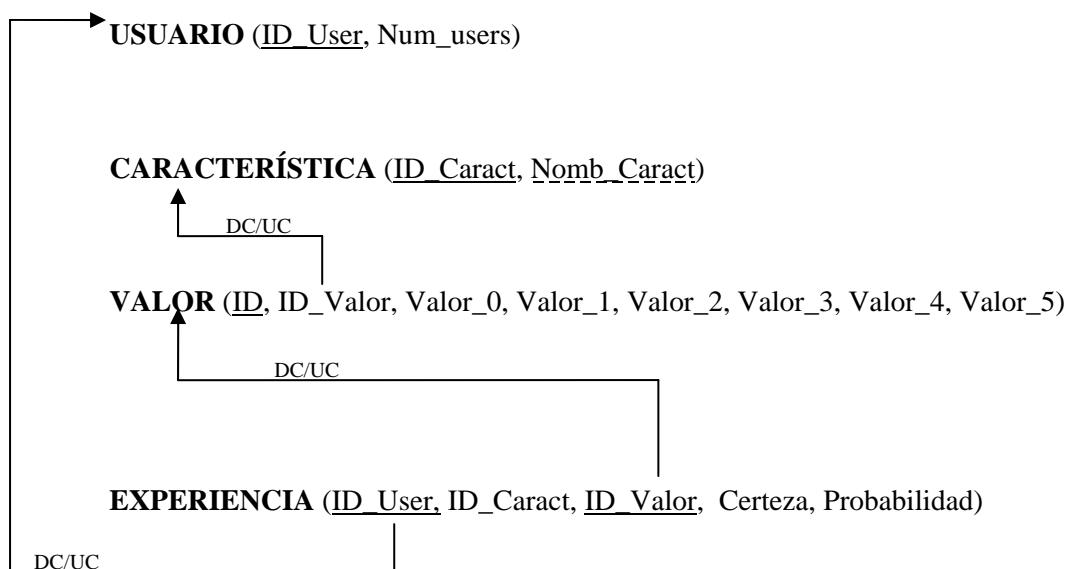


Ilustración 45. Grafo relacional del Entorno de trabajo con modelos de usuario

Para una mejor comprensión del grafo por parte de lector ahora pasaremos a explicar algunas notaciones empleadas en dicho grafo:

- El único modo de borrado y actualización empleado es el DC/UC (*Delete Cascade / Update Cascade*), es decir, que a la hora de efectuar borrados y modificaciones en la tabla principal estas acciones también se llevarán a cabo en la tabla secundaria.
- Las claves ajenas (*Foreign Keys*) no se han especificado en el diagrama para no complicar su legibilidad, por lo que serán especificadas a continuación:
 - La relación VALOR cuenta con una clave ajena, que recibe de la entidad CARACTERÍSTICA.
 - La relación EXPERIENCIA posee tres claves ajenas, que provienen de las tres entidades que relaciona: USUARIO, CARACTERÍSTICA y VALOR.

Una vez comprendida la notación empleada en el grafo pasaremos a analizar y explicar cada una de las decisiones tomadas en la transformación.

Como se comenta en la introducción a esta transformación cada entidad del modelo E/R ha sido transformada en una relación del modelo relacional, sin complicación alguna. En la misma línea han sido transformados los atributos de cada entidad existente.

Ahora bien, en el caso de las interrelaciones ha habido que tomar decisiones importantes a la hora de transformar la interrelación que compone el modelo E/R: para representar la interrelación ha sido necesaria la creación de una nueva relación: EXPERIENCIA, debido a que no es posible representar dicha interrelación mediante relaciones binarias. Entonces la tabla EXPERIENCIA contiene las claves primarias de las dos relaciones que participan con cardinalidad máxima N, que son USUARIO y VALOR, y los atributos de la interrelación, que son la certeza y la probabilidad. Dado que la clave primaria de esta nueva entidad no es más que la concatenación de las claves primarias de las relaciones comentadas, dicha clave primaria está compuesta por claves ajenas. Así pues, la actualización y el borrado de datos se realizarán en cascada.

Sin embargo, con la creación de la tabla EXPERIENCIA comentada no quedan reflejados todos los casos posibles, ya que puede haber valores que se correspondan a características, sin que tengan que estar obligatoriamente relacionados con un usuario. Por ello será necesario añadir en el esquema E/R una interrelación binaria entre CARACTERÍSTICA y VALOR, tal y como se puede observar en la siguiente ilustración:

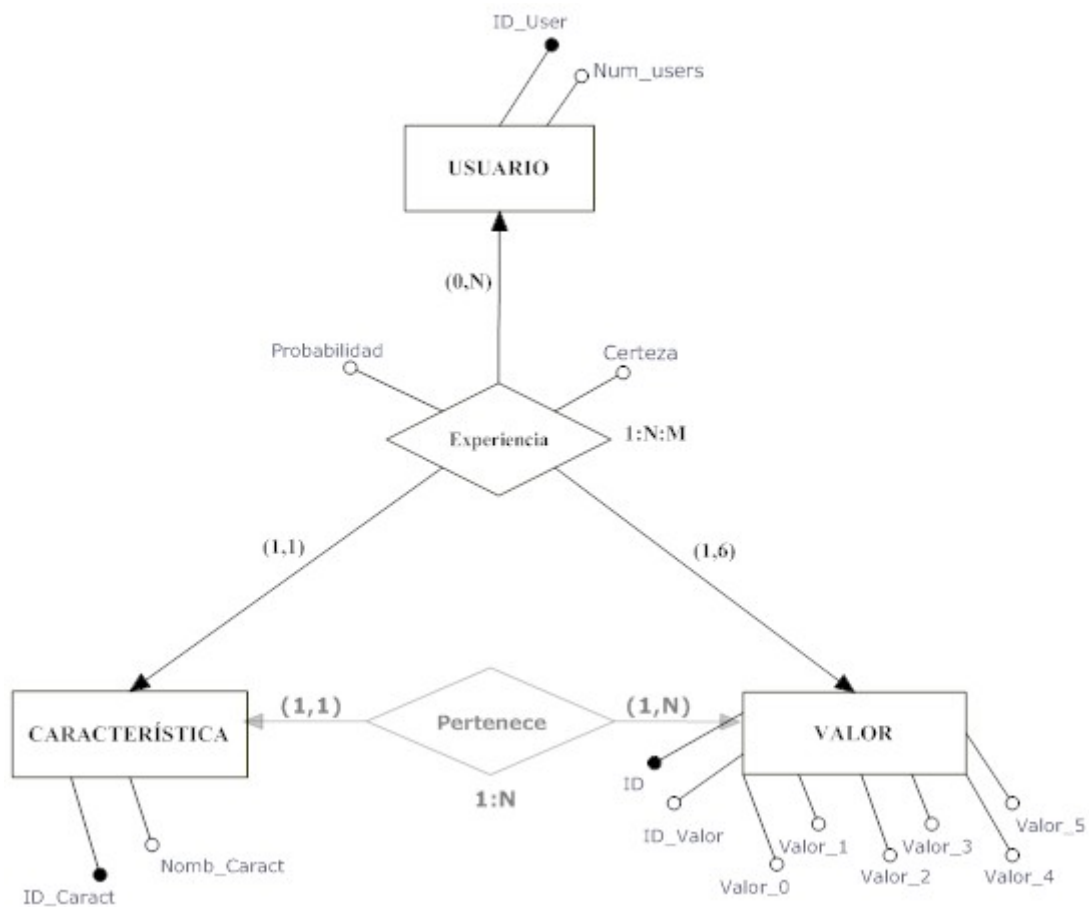


Ilustración 46. Modelo E/R del Entorno de trabajo para modelos de usuario, preparado para la transformación al relacional

A continuación se presenta el grafo relacional que representa la nueva relación 'Corresponde', fruto de la existencia de características y sus valores correspondientes que no están relacionados con ningún usuario.

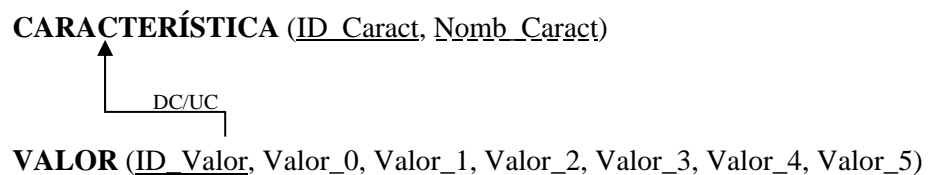


Ilustración 47. Grafo relacional que representa la interrelación 'Corresponde'

No es necesario crear una nueva tabla para representar la relación, porque, tal y como se comentó al inicio de este capítulo, se puede transformar propagando la clave, dando lugar a la creación de una clave ajena en la relación VALOR. El borrado y la actualización de esta clave ajena será en cascada, es decir, que la eliminación de una tupla de la relación CARACTERÍSTICA implica no tener ninguna referencia suya en la relación VALOR y que la modificación de una tupla en esa relación también será modificada en la relación VALOR.

PRUEBAS

¿ Todo correcto ?

6. PRUEBAS

Los progresos realizados en el sistema deben ser medidos o evaluados para conocer las deficiencias y problemas que éste presenta.

Se realizarán pruebas para comprobar que el sistema cumple con todos los requisitos que ha marcado el usuario. Son necesarias para que el usuario valide el sistema, antes de la puesta en explotación.

Se debe insistir sobre todo en probar que el sistema cumple los criterios de aceptación del sistema que sirven de base para asegurar que satisface los requisitos exigidos. Los puntos de atención dependen de la naturaleza del sistema y los requisitos marcados por el usuario.

6.1. WEB CRAWLER

6.1.1. Ejecución de las pruebas

A continuación se presenta una tabla resumen que contiene los resultados de ejecución de todas y cada una de las pruebas definidas para este subsistema (definidas anteriormente en el apartado de Pruebas de validación):

ID	Nombre	Resultado
PR-001	Descargar el contenido de una página Web	Satisfactorio
PR-002	Cambio de contexto correcto	Satisfactorio
PR-003	Clasificar una página por su contenido	Satisfactorio
PR-004	Obtener información relevante de una página	Satisfactorio
PR-005	Obtener los enlaces de una página	Satisfactorio
PR-006	No procesar una página Web ya procesada	Satisfactorio
PR-007	Procesar todos los enlaces de una página	Satisfactorio
PR-008	Almacenar los datos obtenidos de una página Web con información relevante	Satisfactorio
PR-009	Acceso a una página Web fuera del contexto de la aplicación	Satisfactorio

Tabla 14. Ejecución de las pruebas del WebCrawler

6.1.2. Acciones correctivas

No aplica.

6.2. ENTORNO DE TRABAJO CON MODELOS DE USUARIO

6.2.1. Ejecución de las pruebas

Seguidamente se presenta una tabla resumen que contiene los resultados de ejecución de todas y cada una de las pruebas definidas para este subsistema (definidas anteriormente en el apartado de Pruebas de validación):

ID	Nombre	Resultado
PR-001	Obtención de un átomo de información del usuario actual	Satisfactorio
PR-002	Guardar la información obtenida del usuario actual	Satisfactorio
PR-003	Inferir un átomo de información del usuario actual	Satisfactorio
PR-004	Inferir varios átomos de información del usuario actual	Satisfactorio
PR-005	Comprobar un átomo de información del usuario actual	Satisfactorio
PR-006	Almacenar datos inferidos	Satisfactorio
PR-007	Aprender usuario y cerrar sesión	Satisfactorio
PR-008	Entrenamiento del modelo con todos los registros	Satisfactorio
PR-009	Entrenamiento del modelo con los registros seleccionados	Satisfactorio
PR-010	Llamadas a los métodos del modelo	Satisfactorio
PR-011	Evaluación de usuarios dentro del corpus	Satisfactorio
PR-012	Evaluación de usuarios fuera del corpus	Satisfactorio
PR-013	Evaluar varios usuarios	Satisfactorio
PR-014	Carga de la configuración	Satisfactorio
PR-015	Guardar la configuración	Satisfactorio

Tabla 15. Ejecución de las pruebas del Entorno de trabajo con modelos de usuario

6.2.2. Acciones correctivas

No aplica.

CONCLUSIONES

El final del camino

7. CONCLUSIONES

Después de varios meses realizando trabajos de investigación, aprendizaje, análisis, diseño, implementación, etc. se ha conseguido llevar a cabo con éxito este proyecto.

Para lograrlo ha sido necesario recordar conceptos y fundamentos de Java, así como de modelado de Bases de Datos. Gracias a la gran cantidad de librerías proporcionadas por dicho lenguaje de programación se han podido cubrir todos los requisitos del proyecto.

Además, gracias al desarrollo del proyecto de fin de carrera de la Ingeniería Informática Técnica de Gestión, Evaluación de modelos de usuario sobre Oracle, se tenían conocimientos previos acerca del modelado de usuarios, lo cual ha hecho que la comprensión y desarrollo de esa parte del proyecto haya sido menos complicada.

Ahora se puede decir que se han conseguido los objetivos perseguidos desde los inicios del proyecto:

- Se ha conseguido desarrollar una araña Web capaz de extraer información de manera masiva de la red social Facebook.
- La información obtenida por la araña Web puede introducirse en un modelo de usuario definido igualmente, de forma masiva.
- Paralelamente se ha desarrollado una interfaz que permite trabajar con modelos de usuario: capaz de adquirir nueva información de usuarios (a mano o intensivamente) e inferir datos teniendo en cuenta el contenido del modelo. Además, también se puede aprender en el modelo la información inferida.
- Como valor añadido cabe destacar que la interfaz desarrollada es adaptable a distintas tecnologías, esto es, a distintos tipos de modelos de usuarios.

Por todo ello el proyecto puede resultar muy útil para las personas que tengan que trabajar con modelos de usuarios, ya que tiene como funcionalidad principal conocer información desconocida hasta el momento a partir de otros datos sobre el usuario o grupos de usuarios, ampliando así el conocimiento. Esta ampliación podría hacerse de manera aleatoria, pero a través de este sistema desarrollado la nueva información resulta muy fiable.

Lo que es más, permite trabajar con distintos modelos de usuarios, lo que hace que el sistema sea completamente portable. La única restricción es que los modelos de usuarios estén implementados en Oracle o Java.

La aplicación desarrollada resulta cuanto menos útil (por no decir, imprescindible) a la hora de crear nuevo software o, incluso extrapolándolo al ámbito del marketing, en el momento de crear un nuevo producto, ya sea alimenticio, de ropa, etc. puesto que esta aplicación permitirá conocer casi exactamente el/los destinatario/s del nuevo desarrollo, pudiendo realizar una mayor especialización y, por tanto, sobresalir sobre la competencia.

Como valor formativo para el equipo de desarrollo se ha de mencionar la profundización en el modelado de usuarios.

LÍNEAS FUTURAS

La verdad está ahí afuera

8. LÍNEAS FUTURAS

Una vez terminada la implementación del presente proyecto llega la hora de efectuar una pausa y evaluar los posibles desarrollos que se podrían efectuar más adelante, teniendo como base este sistema, o bien las posibles mejoras en el mismo, ya que se trata de un sistema completo, pero no perfecto. A continuación serán comentadas las reflexiones sobre los trabajos que se podrían realizar sobre el sistema:

- Comenzaremos con ‘lo que se ve’, es decir, la **interfaz** del programa. La interfaz actual del programa ha sido concebida para ser lo más amigable e intuitiva posible, aún así, se podría mejorar el aspecto gráfico. Dadas las posibilidades ofrecidas por el lenguaje en que se ha desarrollado, Java, se podría aplicar una interfaz más vistosa y menos ‘dura’ a la vista (como podría ser la combinación de tonos grises y azules que presenta ahora).
- La extracción de información para el modelo de usuarios a través del **Web Crawler** sería posible **aplicar a otras redes sociales**, no sólo a Facebook, ya que el resto de redes sociales también contienen información útil de sus usuarios.
- **Integración del Web Crawler** en el entorno de trabajo para modelos de usuario, de tal forma, que no sea necesario ejecutarlos por separado.
- **Monitorización de los modelos de usuario** que incluye opciones como modificar los modelos de usuario omitiendo un grupo de usuarios, añadiendo una columna que permita decidir si un grupo está activo o inactivo, visualizar las propiedades y características de un grupo de usuarios, etc.
- En la definición del modelo añadir opciones que permitan elegir entre distintos **algoritmos de inferencia** sobre el modelo que se debe aplicar. De esta forma, el usuario podrá elegir el que más se adecue a sus datos.

BIBLIOGRAFÍA

A continuación se detallan todos los recursos de información utilizados para la elaboración del proyecto, clasificados por tipo y, dentro de cada tipo, ordenados por orden de aparición en este documento:

► **Referencias:**

- [Kobayashi, 2000] Kobayashi, M. and Takeda, K. *Information retrieval on the web*. ACM Computing Surveys (ACM Press). 2000.
- [Boldi, 2004] Boldi, Paolo; Massimo Santini, Sebastiano Vigna. *Do Your Worst to Make the Best: Paradoxical Effects in PageRank Incremental Computations*. 2004.
- [Montero, 2009] Montero, José. Recuperación y organización de la información. Universidad Carlos III de Madrid. 2009.
- [Boyd-Ellison, 2007] Boyd, D. M. – Ellison, N. B. *Social network sites: Definition, history, and scholarship*. *Journal of Computer-Mediated Communication*, 2007.
- [Medina-García, 2002] Medina Medina, Nuria – García Cabrera, Lina – Rodríguez Fortiz, M^a José – Parets Llorca, José. *Adaptación al Usuario en Sistemas Hipermedia: El Modelo SEM-HP*. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada y Departamento de Informática, Universidad de Jaén. 2002.
- [Kules, 2000] Kules, Hill. *User Modeling for Adaptive and Adaptable Software Systems*. Department of Computer Science, University of Maryland, USA. 2000.
- [Vialardi, 2006] Vialardi, César. *Aplicación de técnicas de minería de datos para la creación y mantenimiento del modelo del estudiante en sistemas e-learning*. Proyecto de postgrado en la Universidad Autónoma de Madrid. 2006.

- [Rodríguez-Arano, 1999] Rodríguez, Mónica – Arano, Silvia. *Interacción hombre-computadora y diseño de interfaz – Modelado de usuarios en el diseño de interfaces hombre-computadora*. Centro regional de nuevas tecnologías de información. 1999.
- [Fischer, 2000] Fischer, Gerard. *User Modeling in Human-Computer Interaction*. Department of Computer Science and Institute of Cognitive Science, University of Colorado at Boulder, Boulder, U.S.A. 2000.
- [Robertson, 1997] Robertson, S. E. *Theories and models in information retrieval* Journal of documentation. 1977.
- [Belkin-Croft, 1987] Belkin, N. – Croft, W. B. *Retrieval Techniques*. 1987.
- [Belkin-Oddy, 1982] Belkin, N. – Oddy, R. – Brooks, H. *Ask for Information Retrieval: 1; Ask for Information Retrieval: 2*. 1982.
- [Rich, 1979] Rich, Elaine. *User Modelling via stereotypes*. Cognitive Science. 1979.
- [Monden, 1983] Monden, Y., *Toyota Production System: Practical Approach to Management*. Industrial Engineering and Management Press. Norcross. 1983.
- [Reitman-M. Olson, 1995] Reitman Olson, Judith – M. Olson, Gary, *The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS*, en: R. M. Baecker, J. – Grudin, W. – A. S. Buxton, S. Greenberg: *Readings in Human-Computer Interaction: Towards the Year 2000*. 1995.
- [Chomsky, 1956] Chomsky, Noam. *Three Models for the Description of Language*. IRE Transactions on Information Theory. 1956.
- [Chomsky, 1957] Chomsky, Noam. *Syntactic Structures*, Mouton, The Hague. 1957.
- [Naranjo, 2003] Naranjo Vélez, Edilma. *Seminario Estudios de Usuarios*. Escuela Interamericana de Bibliotecología, Universidad de Antioquia. 2003.
- [Gaudioso, 2002] Gaudioso Vázquez, Elena. *Tesis Doctoral: Contribuciones al Modelado de Usuario en Entornos Adaptativos de Aprendizaje y Colaboración a través de Internet mediante técnicas de Aprendizaje Automático*. Dpto. de Inteligencia Artificial, Universidad Nacional de Educación a Distancia. 2002.

- [Stephanidis, 1998] Stephanidis, C. – Paramythis, A – Sfyra, M. – Stergiou, A. – Maou, N. – Leventis, G. – Kargiandidis, C. *Adaptable and adaptive user interfaces for disabled users in avanti project*. En *Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services*. New York. 1998.
- [Ayçaguer, 2001] Ayçaguer LC, Silvia – A., Benavides. *El enfoque bayesiano: otra manera de inferir*. Gaceta sanitaria. 2001.
- [Burton-Brown,1982] Burton, R. R. – Brown, J. S. *An investigation of computer coaching for informal learning activities in: Sleeman, D. H. – Brown, J. S. Intelligent Tutoring Systems, Academic Press London-New York*. 1982.
- [P.P. da Cruz, 2003] P.P. da Cruz, Rui Alexandre – García Peñalvo, Francisco J. – Alonso Romero, Luis. *Informe técnico – Perfiles de Usuario: en la senda de la personalización*. Dpto. de Informática y Automática, Universidad de Salamanca. 2003.
- [CMR, 1987] *The Computer Museum Report – Smart Machines*. 1987.
- [Encarnaçao, 1999] Encarnaçao, L. Miguel – Stoev, Stanislav L. *An Application-Independent Intelligent User Support System Exploiting Action-Sequence Based User Modelling*. 1999.
- [Kobsa, 1994] Kobsa, Alfred – Müller, Dietmar – Nill, Andreas. *KN-AHS: An Adaptive Hypertext Client of the User Modelling System BGP-MS*. Dept. of Information Science, Univ. of Konstanz, Germany. 1994.
- [Kobsa, 1995] Kobsa, Alfred – Pohl, Wolfgang. *User Modelling and User-Adapted Interaction*. Dept. of Information Science, Univ. of Konstanz, Germany. 1995.
- [Fischer, 1985] Fischer, Gerhard – Lemke, A. C. – Schwab, T. *Knowledge-based help systems in: Borman, L. – Curtis, B. Proceedings of CH'85 Conference on Human Factors in Computing Systems*, ACM, New York. 1985.
- [McKeown, 1990] McKeown, K. R. *User modelling and user interfaces*. In: *Proceedings of AAAI-90 Eighth National Conference on Artificial Intelligence*, AAAI Press / The MIT Press, Cambridge, MA. 1990.

- [Horvitz, 1998] Horvitz, Eric – Breese, Jack – Heckerman, David – Hovel, David – Romelse, Koos. *The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison. 1998.
- [Hípola, 1997] Hípola, Pedro – Senso, José A. *Sistemas Push: información a la carta*. 1997.
- [Gaudioso, 2002] Gaudioso, Elena – G. Boticario, Jesús. *WebDL: Un sistema adaptativo para el aprendizaje cooperativo a distancia a través de Internet*. Dpto. de Inteligencia Artificial, Universidad Nacional de Educación a Distancia. 2002.
- [Mitchell, 1990] Mitchell, T.M. – Allen, J. – Chalasani, P. – Cheng, J. – Etzioni, O. – Ringuette, M. – Schlimmer, J.C. *Theo: A framework for self-improving systems*. In K. VanLehn, editor, *Architectures for Intelligence*. Erlbaum, Hillsdale. 1990.
- [Corcos, 2003] Corcos, Daniel. *El modelo Espiral*. 2003.
- [ISI, 2002] Apuntes de la asignatura *Ingeniería del Software I*. Ingeniería Técnica en Informática de Gestión, Universidad Carlos III de Madrid. 2002.
- [ISII, 2003] Apuntes de la asignatura *Ingeniería del Software II*. Ingeniería Técnica en Informática de Gestión, Universidad Carlos III de Madrid. 2003.
- [FBD, 2004] Apuntes de la asignatura *Ficheros y Bases de Datos*. Ingeniería Técnica en Informática de Gestión, Universidad Carlos III de Madrid. 2004.
- [DBD, 2005] Apuntes de la asignatura *Diseño de Bases de Datos*. Ingeniería Técnica en Informática de Gestión, Universidad Carlos III de Madrid. 2005.
- [Eckel, 1998] Eckel, Bruce. *Thinking in Java*. 1998.

► Enlaces:

- Ranking Alexa: <http://www.rankingalexa.com/>
http://www-static.cc.gatech.edu/classes/cs6751_97_winter/Topics/user-model/
- *The InterBook Home Page*,
<http://www.contrib.andrew.cmu.edu/~plb/InterBook.html>
- Horvitz, Eric. *Adaptive Systems & Interaction*. Microsoft Research,
<http://research.microsoft.com/~horvitz/lum.htm>
- Horvitz, Eric – Breese, Jack – Heckerman, David – Hovel, David – Rommelse, Koos. *Decision Theory & Adaptive Systems Group*. Microsoft Research,
<http://research.microsoft.com/~horvitz/lumiere.HTM?0sr=a>
- Tutorial de Java:
<http://www.itapizaco.edu.mx/paginas/JavaTut/froufe/introduccion/indice.html>
- Tutorial de Java:
<http://www.cica.es/formacion/JavaTut/Intro/tabla.html>
- Tutor básico de Java, adaptación de Sun:
<http://ji.ehu.es/LMAlonso/SW/java/Bib/tutorjava/html/basico.html>
- Base de datos Oracle:
<http://www.compiere.com.ar/support/install/installOracle.html#detail>

ANEXOS

Todo lo demás

ANEXO A. Aplicación Web Crawler

Para comenzar se detallará el catálogo de los requisitos que están encargados de cubrir la completa funcionalidad de la aplicación requeridas por el usuario final. Los requisitos de usuarios pueden ser de capacidad y de restricción.

Después se van a definir los requisitos de software, que son una referencia para verificar el diseño y el producto, pues ambos deben cubrir los requisitos que se hayan marcado.

Finalmente se detallarán las pruebas de validación efectuadas para comprobar que el sistema cumple con todos los requisitos que ha marcado el usuario.

Requisitos de usuario

RCP-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Leer el contenido de una página Web de <i>Facebook</i> e interpretar la información incluida en ella

RCP-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Distinguir las páginas con información relevante de las que sólo contienen enlaces a otras páginas

RCP-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Conocer las páginas Web ya visitadas para evitar procesar más de una vez la misma página

RCP-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Obtener la siguiente información desde la página Web del perfil público de un usuario: páginas de las que es admirador, número de páginas de las que es fan y número de amigos

RCP-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar la información obtenida del perfil de usuario en una base de datos

RCP-006	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Iniciar la aplicación desde un enlace a una página Web de <i>Facebook</i> pasado como parámetro o desde un estado anterior

RCP-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Extraer los enlaces a otras páginas de <i>Facebook</i> de la página que se esté procesando

RCP-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Acceder a todas las posibles páginas de <i>Facebook</i> , a través de los enlaces obtenidos desde otras páginas.

RRS-001	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las páginas de <i>Facebook</i> accedidas estarán en castellano

RRS-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El tiempo de respuesta de la base de datos no será superior a un segundo

RRS-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Todas las páginas Web de <i>Facebook</i> con información de perfil de un usuario tienen el mismo formato

RRS-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Todas las páginas Web de <i>Facebook</i> con información no relevante, pero con enlaces a otras páginas, tienen el mismo formato

RRS-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La base de datos empleada para almacenar la información extraída será Oracle

RRS-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las listas de Webs visitadas y pendientes de visitar se almacenarán en la base de datos

RRS-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El contexto de las direcciones de las páginas Web es: http://www.facebook.es/

RRS-008	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las direcciones de páginas Web almacenadas sólo contendrán la parte cambiante, es decir, no se almacenará la parte del contexto de las mismas

RRS-009	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las tablas que contienen la información extraída de los perfiles tendrán 10 millones de registros como máximo

Requisitos de software

RFN-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación será capaz de procesar páginas Web, dentro del contexto de http://www.facebook.es/ , accediendo a su contenido
Nombre	Procesamiento de páginas Web dentro de contexto

RFN-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Reconocimiento del contenido de una página Web procesada, clasificándola como una página con información relevante o con enlaces a otras páginas
Nombre	Clasificación de las páginas Web procesadas

RFN-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dada una página Web con enlaces a otras páginas, se extraerán todos los enlaces a otras páginas
Nombre	Extracción de los enlaces contenidos en una página Web

RFN-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dada una página Web que contenga información relevante, se extraerán los siguientes datos del perfil de usuario contenido en ella: número de páginas de las que es admirador, número de páginas de las que es fan, número de amigos y, como máximo, 20 páginas de las que es admirador, divididas en 4 categorías
Nombre	Extracción de información relevante contenida en una página Web

RFN-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los datos extraídos de una página con información relevante en una base de datos
Nombre	Almacenamiento de la información relevante en base de datos

RFN-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Inicio de la ejecución del Crawler a partir de una página Web, dentro del contexto http://www.facebook.es/ , pasada como parámetro
Nombre	Ejecución de la aplicación partiendo de una dirección Web pasada como parámetro

RFN-007	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Inicio de la ejecución del Crawler a partir de un estado anterior, es decir, partiendo de la información contenida en las listas de páginas visitadas y pendientes de visitar
Nombre	Ejecución de la aplicación partiendo de un estado anterior

RFN-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mantenimiento de dos listas que contengan direcciones de páginas Web: una con las que ya se ha visitado y otra con las que están pendientes de visitar
Nombre	Mantenimiento de listas con Webs pendientes de visitar y ya visitadas

RFN-009	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los datos extraídos de una página con enlaces en dos listas: una con páginas pendientes de visitar y otra con páginas visitadas
Nombre	Almacenamiento de enlaces extraídos de una página

RFN-010	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Procesar las páginas cuyos enlaces se encuentran en la lista que contiene los enlaces pendientes de visitar
Nombre	Procesamiento de páginas pendientes de visitar

RRD-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación se ejecutará 24 horas al día
Nombre	Aplicación disponible 24 horas al día

RIN-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El programa mostrará una traza con la dirección Web de las páginas que va procesando
Nombre	Trazas de direcciones de páginas Web procesadas

RIN-002	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En caso de error el programa mostrará el mensaje correspondiente a dicho error
Nombre	Mensajes de error

ROP-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Inicialización de las listas de direcciones de páginas Web visitadas y pendientes de visitar, insertando la dirección pasada como parámetro, en el caso de que se pase una dirección como parámetro
Nombre	Inicialización de las listas de Webs visitadas y pendientes de visitar

ROP-002	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Si el contexto de la dirección Web pasada como parámetro es distinto de http://www.facebook.es/ , deberá cambiarse el contexto del parámetro al especificado
Nombre	Contexto por defecto del parámetro de inicio

ROP-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dado un enlace a una página Web, extraído de la página Web que se esté procesando, se añadirá a la lista de páginas Web pendientes de visitar si no se encuentra ya en dicha lista
Nombre	Añadir un enlace a la lista de enlaces pendientes de visitar si no estaba previamente

ROP-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Si el contexto de un enlace a una dirección Web es distinto de <u>http://www.facebook.es/</u> , deberá cambiarse el contexto dicho enlace al especificado
Nombre	Contexto por defecto en enlaces

ROP-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dado un enlace a una página Web, extraído de la página Web que se esté procesando, se añadirá a la lista de páginas Web pendientes de visitar si el contexto de dicho enlace es: <u>http://www.facebook.es/directory/people/</u> o bien <u>http://www.facebook.es/peopole/</u>
Nombre	Contextos válidos de los enlaces

ROP-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dado un enlace a una página Web se accederá a la misma para obtener su contenido íntegro
Nombre	Descarga de contenidos de páginas Web

ROP-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dado un enlace a una página Web extraído de la lista de páginas pendientes de visitar, habrá que añadirle el contexto http://www.facebook.es/ para poder descargar su contenido
Nombre	Añadir contexto a enlace extraído de lista de pendientes de visitar

ROP-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dado el contenido de una página Web se descargará la información contenida en la misma si contiene la palabra “ <i>admirador</i> ”
Nombre	Descarga de información de una página Web

ROP-009	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dado el contenido de una página Web se descargará la lista de enlaces contenida en la misma si no contiene la palabra “ <i>admirador</i> ”
Nombre	Descarga de enlaces de una página Web

ROP-010	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Al procesar una enlace a una página Web se insertará automáticamente en la lista de páginas Web ya visitadas
Nombre	Procesamiento de enlaces

RRC-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El tamaño máximo de las tablas que contienen la información extraída de los perfiles será de 10 millones de registros
Nombre	Tamaño máximo de tablas

RCM-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se comprobará que el parámetro pasado al inicio del programa se trata de una dirección Web válida
Nombre	Comprobación de parámetros

RCM-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El nombre de las tablas de la base de datos tendrán el siguiente formato: "table_" + <número_tabla>. El campo número_tabla empieza en 0 y se irá incrementando de uno en uno
Nombre	Nombres de tablas en la base de datos

RDC-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El lenguaje empleado para la documentación del sistema será el castellano
Nombre	Lenguaje castellano

RDC-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se elaborará un manual de usuario de la aplicación
Nombre	Manual de usuario

RDC-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Los comentarios deberán describir todas las funciones del código fuente de manera clara y concisa
Nombre	Comentarios del código fuente

RDC-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El tipo letra empleado para toda la documentación del proyecto es Times New Roman
Nombre	Tipo de letra en la documentación

RDC-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El tamaño letra empleado para toda la documentación del proyecto es de 11 puntos
Nombre	Tamaño de letra en la documentación

RDC-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En la redacción del manual de usuario se emplearán ejemplos de uso, evitando la utilización de tecnicismos que no sean de dominio público. En el caso de que fuera imprescindible su uso habría que explicar su significado en el mismo manual.
Nombre	Legibilidad del manual de usuario

RSG-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenamiento del estado de la aplicación en una base de datos, para evitar pérdidas de información debido a cortes de luz
Nombre	Almacenamiento del estado del sistema

Pruebas de validación

El campo ID hace referencia al identificador único de la prueba. El formato a seguir es el siguiente: PR-<números>, donde el campo <números> son números enteros consecutivos incrementados en una unidad.

PR-001	
Nombre	Descargar el contenido de una página Web
Descripción	Comprobar que se accede a una página Web y se descarga su contenido íntegro
Elementos relacionados	RCP-001, RCP-002, RRS-001, RRS-007
Especificaciones de entrada	Acceder a distintas páginas Web dentro del contexto de la aplicación
Especificaciones de salida	El sistema obtendrá el contenido de cada página Web en un String
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando

PR-002	
Nombre	Cambio de contexto correcto
Descripción	Verificar que dada una dirección Web de <i>Facebook</i> fuera de contexto, se cambia de contexto sin que varíe el contenido de la página más que en el idioma
Elementos relacionados	RRS-007
Especificaciones de entrada	Dirección de página Web de <i>Facebook</i> fuera del contexto http://www.facebook.es/
Especificaciones de salida	Dirección de página Web de <i>Facebook</i> dentro del contexto descrito
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando

PR-003	
Nombre	Clasificar una página por su contenido
Descripción	Determinar si una página Web contiene información relevante o enlaces a otras páginas Web
Elementos relacionados	RCP-001, RCP-002, RRS-001, RRS-003, RRS-004, RRS-007
Especificaciones de entrada	Acceder a distintas páginas Web dentro del contexto de la aplicación
Especificaciones de salida	Se mostrará el literal “perfil” si la página contiene información relevante o el literal “enlaces” si contiene enlaces a otras páginas Web
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando

PR-004	
Nombre	Obtener información relevante de una página
Descripción	Comprobar que la información descargada de la página Web dentro del contexto (número de páginas de las que es admirador, número de páginas de las que es fan, número de amigos y las páginas de las que es admirador) es la misma que la que contiene la Web efectivamente
Elementos relacionados	RCP-004, RRS-001, RRS-003, RRS-007, PR-003
Especificaciones de entrada	Acceder a distintas páginas Web dentro del contexto de la aplicación que hayan mostrado el literal “perfil” en la prueba PR-003
Especificaciones de salida	Toda la información extraída de la página Web será mostrada por pantalla y debe coincidir con la contenida en la Web
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando y al menos un navegador Web instalado en el mismo

PR-005	
Nombre	Obtener los enlaces de una página
Descripción	Probar que las direcciones Web de los enlaces obtenidos de la página Web dentro del contexto son los mismos que los que contiene la Web efectivamente
Elementos relacionados	RCP-007, RRS-001, RRS-004, RRS-007, PR-003
Especificaciones de entrada	Acceder a distintas páginas Web dentro del contexto de la aplicación que hayan mostrado el literal “enlaces” en la prueba PR-003
Especificaciones de salida	Todos los enlaces extraídos de la página Web serán mostrados por pantalla y deben coincidir con los contenidos en la Web
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando y al menos un navegador Web instalado en el mismo

PR-006	
Nombre	No procesar una página Web ya procesada
Descripción	Verificar que no se procesa dos veces la misma página Web
Elementos relacionados	RCP-001, RCP-003, RCP-008, RRS-004, RRS-007
Especificaciones de entrada	Acceder a la misma página Web dentro del contexto de la aplicación al menos dos veces
Especificaciones de salida	La segunda vez que se intente acceder a la página Web no se procesará dicha página
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando

PR-007	
Nombre	Procesar todos los enlaces de una página
Descripción	Comprobar que, una vez extraídos todos los enlaces de una página, el sistema procesa todas las páginas Web de esos enlaces
Elementos relacionados	RCP-008, RRS-001, RRS-004, RRS-007, PR-003
Especificaciones de entrada	Acceder a una página Web dentro del contexto de la aplicación que haya mostrado el literal “enlaces” en la prueba PR-003
Especificaciones de salida	Mostrar por pantalla la dirección Web de cada enlace procesado
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando y al menos un navegador Web instalado en el mismo

PR-008	
Nombre	Almacenar los datos obtenidos de una página Web con información relevante
Descripción	Comprobar que se almacenan de manera correcta todos los datos extraídos de una página Web con información relevante
Elementos relacionados	RCP-004, RCP-005, RRS-001, RRS-002, RRS-003, RRS-005, RRS-007, PR-003
Especificaciones de entrada	Acceder a distintas páginas Web dentro del contexto de la aplicación que hayan mostrado el literal “perfil” en la prueba PR-003
Especificaciones de salida	Creación de nuevas tuplas en la base de datos correspondiente con la información extraída de la página Web
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando y que tenga la base de datos del sistema instalada

PR-009	
Nombre	Acceso a una página Web fuera del contexto de la aplicación
Descripción	Se intentará descargar el contenido de una página Web fuera del contexto de la aplicación
Elementos relacionados	RCP-006, RRS-007
Especificaciones de entrada	Acceder a distintas páginas Web fuera del contexto de la aplicación
Especificaciones de salida	Se mostrará una línea con el mensaje de error correspondiente y se seguirá con la ejecución del programa
Entorno de prueba	Conexión a Internet en el equipo que se esté ejecutando

Casos de uso

A continuación se muestran las descripciones textuales de los casos del uso de la aplicación Web Crawler:

CU-001	
Nombre	Descarga de contenido de una página
Descripción	Obtención del contenido íntegro de una página Web dada
Actores	Robot
Precondiciones	Que la página Web dada exista, sea accesible y esté dentro del contexto definido para la aplicación
Flujo	<ol style="list-style-type: none">1. Abrir conexión URL a la dirección Web de la página2. Repetir<ol style="list-style-type: none">i. Leer líneaii. Añadir línea a la cadena de salida3. Hasta la longitud de la página
Postcondiciones	Cadena de caracteres con el contenido íntegro de la página Web

CU-002	
Nombre	Ir al enlace siguiente
Descripción	Acceder a la lista de enlaces no visitados y obtener el primer elemento de la misma
Actores	Robot
Precondiciones	Que la lista de enlaces no visitados no esté vacía
Flujo	<ol style="list-style-type: none">1. Realizar consulta sobre la base de datos, sobre la lista de enlaces no visitados2. Devolver el primer elemento del resultado de la consulta
Postcondiciones	Cadena de caracteres con la dirección Web que había almacenada en la primera posición de la lista de enlaces no visitados

CU-003	
Nombre	Obtener información de la página Web
Descripción	Recorrer el contenido de una página Web y obtener la información relevante de dicha Web: número de páginas de las que es fan, número de amigos y páginas de las que es admirador
Actores	Robot
Precondiciones	Disponer de la cadena de caracteres con el contenido de la página
Flujo	<ol style="list-style-type: none"> 1. Guardar la información dispuesta en la posición correspondiente: <ol style="list-style-type: none"> i. al nombre de las categorías de las páginas de las que es admirador ii. al nombre de las páginas de las que es fan iii. al número de amigos iv. al número de páginas de las que es fan
Postcondiciones	Enteros con el número de amigos y de páginas de las que es fan y tabla con los nombres de las categorías y páginas correspondientes a cada una de ellas

CU-004	
Nombre	Almacenar información de una página Web
Descripción	Guardar en la base de datos la información relevante obtenida de una página Web
Actores	Robot
Precondiciones	Haber obtenido la información de la página Web y que esté abierta la conexión con la base de datos
Flujo	<ol style="list-style-type: none"> 1. Almacenar la información (identificador de usuario, número de páginas de las que es fan, número de amigos y tabla con páginas de las que es admirador) en la tabla correspondiente
Postcondiciones	Inserción de la nueva información en la base de datos

CU-005	
Nombre	Obtener enlaces de una página Web
Descripción	Recorrer el contenido de una página Web y obtener todos los enlaces que haya en la misma
Actores	Robot
Precondiciones	Disponer de la cadena de caracteres con el contenido de la página
Flujo	<ol style="list-style-type: none"> 1. Repetir <ol style="list-style-type: none"> i. Buscar inicio de un enlace ii. Buscar fin de un enlace iii. Eliminar contexto del enlace iv. Obtener la dirección del enlace sin contexto en la lista de enlaces no visitados 2. Hasta llegar al final del contenido de la página
Postcondiciones	Todas las direcciones de páginas Web contenidas en la página procesada

CU-006	
Nombre	Cambiar de contexto una página Web
Descripción	Cambiar el contexto de la dirección de la página Web al contexto por defecto: http://www.facebook.es/
Actores	Robot
Precondiciones	Que la dirección Web tenga un contexto que no pertenezca al contexto por defecto
Flujo	<ol style="list-style-type: none"> 1. Obtener la parte de la dirección Web correspondiente al contexto 2. Compararla con el contexto definido por defecto 3. Si es distinto del contexto por defecto <ol style="list-style-type: none"> i. Quitar la parte del contexto ii. Añadir el contexto por defecto a la dirección
Postcondiciones	Misma página Web, pero cambiando de contexto la dirección

CU-007	
Nombre	Almacenar enlaces
Descripción	Guardar los enlaces extraídos de una página Web en la base de datos
Actores	Robot
Precondiciones	Tener los enlaces sacados de una página Web y que exista conexión con la base de datos
Flujo	<ol style="list-style-type: none"> 1. Repetir <ol style="list-style-type: none"> i. Leer enlace ii. Almacenar dirección en la lista de enlaces pendientes de visitar 2. Hasta llegar al final de los enlaces
Postcondiciones	Nuevos registros en la lista de enlaces de la base de datos

CU-008	
Nombre	Eliminar enlaces
Descripción	Borrar un enlace de la lista de enlaces pendientes de visitar
Actores	Robot
Precondiciones	Que la lista de enlaces pendientes de visitar no esté vacía, que exista conexión con la base de datos y que exista el enlace en la lista
Flujo	<ol style="list-style-type: none"> 1. Localizar el enlace en la lista de enlaces no visitados 2. Realizar el borrado del enlace sobre la base de datos
Postcondiciones	Lista de enlaces pendientes de visitar con un registro menos

ANEXO B. Aplicación Entorno de trabajo con modelos de usuario

Primeramente se detallará el catálogo de los requisitos que están encargados de cubrir la completa funcionalidad de la aplicación requeridas por el usuario final. Los requisitos de usuarios pueden ser de capacidad y de restricción.

Seguidamente se van a definir los requisitos de software, que son una referencia para verificar el diseño y el producto, pues ambos deben cubrir los requisitos que se hayan marcado.

Para finalizar se detallan las pruebas de validación efectuadas para comprobar que el sistema cumple con todos los requisitos que ha marcado el usuario.

Requisitos de usuario

RCP-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Selección del tipo de modelo de usuarios

RCP-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cambios en las propiedades de modelo de usuarios elegido

RCP-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Carga de datos desde el fichero de configuración

RCP-004	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Salvar la configuración en un fichero

RCP-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Adquirir información acerca del usuario actual

RCP-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Obtener información acerca del usuario actual a través de la inferencia del modelo

RCP-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los datos del usuario actual inferidos sobre el modelo, positivamente

RCP-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los datos del usuario actual inferidos sobre el modelo, negativamente

RCP-009	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cerrar la sesión del usuario actual e iniciarla con un nuevo usuario

RCP-010	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Introducir nuevos datos en el modelo a partir de una tabla de una base de datos

RCP-011	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Modificar los atributos de la base de datos de la que se extrae la nueva información para introducirla

RCP-012	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Evaluar las características de usuarios no existentes en la base de datos

RCP-013	
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Evaluar las características de usuarios existentes en la base de datos

RCP-014	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Modificar los atributos de la base de datos desde la que se va a producir la evaluación de los usuarios

RRS-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La base de datos empleada para almacenar la información extraída será Oracle

RRS-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Los tipos de modelos de usuarios son Oracle o Java

RRS-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Sólo se pueden inferir características que se encuentren introducidas en la base de datos elegida

RRS-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Sólo se pueden inferir valores que se encuentren introducidas en la base de datos elegida

RRS-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Al seleccionar el rango de valores de identificadores de usuarios, el límite inferior siempre será menor que el límite superior

RRS-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La evaluación se efectuará sobre, al menos, un usuario

RRS-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La adquisición de información inferida sólo podrá efectuarse en el caso de que se hayan inferido un átomo o varios de información

RRS-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación estará implementada en castellano

Requisitos de software

RFN-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El sistema proporcionará los recursos necesarios para instanciar un modelo de usuario sobre un objeto Java
Nombre	Selección de un modelo de usuario de tipo Java

RFN-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El sistema proporcionará los recursos necesarios para manejar un modelo de usuarios almacenado en una base de datos de tipo Oracle
Nombre	Selección de un modelo de usuario de tipo Oracle

RFN-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El editor leerá los datos contenidos en un fichero de configuración para inicializar los datos configurables
Nombre	Lectura del fichero de configuración

RFN-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El editor modificará el modelo de usuarios si así es solicitado
Nombre	Cambiar el modelo de usuarios seleccionado

RFN-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar en la base de datos la información del usuario actual introducida en la aplicación
Nombre	Obtener información sobre el usuario actual

RFN-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Obtener información del usuario actual no conocida, a través de la información conocida sobre el mismo y de los datos contenidos en el modelo de usuarios
Nombre	Inferir información sobre el usuario actual

RFN-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Calcular una certeza sobre una cierta información del usuario actual, basándose en la información contenida en el modelo y en los datos conocidos sobre el usuario actual
Nombre	Dada una cierta información sobre el usuario actual, obtener una certeza

RFN-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar en el modelo la información inferida elegida, haciendo que ésta forme parte del modelo, con una certeza positiva
Nombre	Adquirir información del usuario actual positivamente

RFN-009	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar en el modelo la información inferida elegida, haciendo que ésta forme parte del modelo, con una certeza negativa
Nombre	Adquirir información del usuario actual negativamente

RFN-010	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los datos de la sesión del usuario actual y crear un nuevo usuario
Nombre	Cerrar la sesión del usuario actual

RFN-011	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Obtener todas los registros de la tabla seleccionada de la base de datos elegida
Nombre	Seleccionar toda la información de una tabla de una base de datos

RFN-012	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Obtener todos los registros de la tabla seleccionada de la base de datos elegida con los identificadores de usuarios comprendidos en el rango introducido
Nombre	Seleccionar un rango de información de una tabla de una base de datos

RFN-013	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Guardar la información adquirida de una tabla de la base de datos en el modelo de usuarios elegido
Nombre	Almacenar la información obtenida de la base de datos en el modelo de usuarios seleccionado

RFN-014	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Modificación del nombre del ordenador al que se conecta, el usuario, la contraseña, el nombre de la base de datos y/o de la tabla de la base de datos de la que se extrae la nueva información para introducirla
Nombre	Cambios en los atributos de la base de datos de la que se extrae información

RFN-015	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Para cada uno de los usuarios elegidos que no existan previamente en la base de datos: aprender todas las características excepto una; entre un aprendizaje y otro, inferir la característica no aprendida, obteniendo distintos valores habiendo aprendido de 0 a n características
Nombre	Evaluación de usuarios no existentes en la base de datos

RFN-016	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Para cada uno de los usuarios elegidos que existan previamente en la base de datos: aprender todas las características excepto una; entre un aprendizaje y otro, inferir la característica no aprendida, obteniendo distintos valores habiendo aprendido de 0 a n características
Nombre	Evaluación de usuarios existentes en la base de datos

RFN-017	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Modificación del nombre del ordenador al que se conecta, el nombre de la base de datos y/o de la tabla de la base de datos desde la que se va a producir la evaluación de los usuarios
Nombre	Cambios en los atributos de la base de datos de la evaluación

RIN-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El editor contendrá una barra de menú
Nombre	Barra de menú

RIN-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El menú incluirá una opción para cerrar la sesión del usuario actual
Nombre	Cerrar sesión en el menú

RIN-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En el menú se incluirá una opción que permita cargar en la aplicación los datos contenidos en el fichero de configuración
Nombre	Carga del fichero de configuración en el menú

RIN-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La interfaz contendrá una pestaña dedicada a la adquisición de información del usuario actual
Nombre	Pestaña Adquisición

RIN-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La interfaz contendrá una pestaña dedicada a la inferencia de nueva información sobre el usuario actual
Nombre	Pestaña Consulta

RIN-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La interfaz contendrá una pestaña dedicada a la adquisición de nuevos datos sobre nuevos usuarios contenidos en una base de datos
Nombre	Pestaña Entrenamiento

RIN-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La interfaz contendrá una pestaña para poder introducir los datos de configuración del modelo de usuarios elegido
Nombre	Pestaña Definición

RIN-008	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La interfaz contendrá una pestaña dedicada a la valoración de la inferencia de las características de los usuarios elegidos
Nombre	Pestaña Evaluación

RIN-009	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las características y sus valores se seleccionarán de una lista desplegable, cargado previamente con los posibles valores de cada una de ellas
Nombre	Selección de características y valores

RIN-010	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las características y valores seleccionados en la pestaña Adquisición se insertarán en una lista situada en la misma pestaña.
Nombre	Insertión de información del usuario actual

RIN-011	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Las inferencias y certezas calculadas por la aplicación en la pestaña Consulta, serán mostradas en una lista situada en dicha pestaña
Nombre	Lista de inferencias y certezas

RIN-012	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Sólo se podrá introducir la información del tipo de modelo de usuarios seleccionado
Nombre	Información del modelo de usuarios

RIN-013	
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En la pestaña donde se definen los parámetros del modelo de usuarios debe existir un botón que permita vaciar la información de dichos parámetros
Nombre	Botón Reiniciar

RIN-014	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En la pestaña de Definición del modelo deberá haber un botón que permita almacenar toda la información contenida en dicha pestaña en un fichero
Nombre	Guardar configuración del modelo en fichero

RIN-015	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En la pestaña Entrenamiento se debe poder seleccionar la tabla completa o bien un rango de usuarios para ser entrenados
Nombre	Selección del rango de usuarios a entrenar

RIN-016	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En la pestaña Consulta debe existir al menos un botón que permita insertar las inferencias efectuadas sobre la característica elegida en la base de datos
Nombre	Inserción de las inferencias

ROP-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se instanciará un objeto del tipo especificado en la configuración del modelo de usuarios
Nombre	Implementación Java del modelo de usuarios

ROP-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se instanciará el método correspondiente, obteniendo el nombre de los datos de configuración del modelo. Una vez instanciado, se invocará
Nombre	Ejecución de métodos con un modelo de usuarios Java

ROP-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Es necesario invocar el método seleccionado, obteniendo su nombre de los datos de configuración del modelo; anteponiendo el nombre del paquete seguido de un punto
Nombre	Ejecución de métodos con un modelo de usuarios Java

ROP-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se almacenará en la base de datos del modelo los datos referentes al usuario actual, tanto obtenidos como inferidos, si es que éstos no han sido guardados con anterioridad. Y se asigna un nuevo usuario para la aplicación
Nombre	Cierre de sesión aprendiendo

ROP-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La selección de los usuarios a evaluar se realiza de manera aleatoria
Nombre	Evaluación de usuarios

ROP-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La elección de la característica que será inferida en la evaluación será aleatoriamente
Nombre	Característica a inferir en la evaluación

RRC-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cada registro de la tabla de valores puede contener hasta 6 datos, pudiendo haber varios registros para una característica dada
Nombre	Número de valores por característica

RRC-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	De un usuario sólo se pueden obtener hasta 4 características, con 5 valores cada una de ellas
Nombre	Obtención de características de un usuario

RCM-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>El contenido del fichero de configuración debe ser el siguiente:</p> <p>Tipo Modelo:</p> <p>{ si Java }</p> <p>[JRE:</p> <p>Clase principal:]</p> <p>{ si Oracle }</p> <p>[Host:</p> <p>Usuario:</p> <p>Contraseña:</p> <p>Nombre del paquete:</p> <p>Nombre de la tabla:]</p> <p>1.Método añadir característica:</p> <p>2.Adquirir un átomo de información:</p> <p>3.Inferir un átomo de información:</p> <p>4.Comprobar un átomo de información:</p> <p>5.Inferir varios átomos de información:</p> <p>6.Cerrar sesión (aprendiendo):</p>
Nombre	Contenido del fichero de configuración

RCM-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El nombre de los campos de las tablas a partir de las que se produce un entrenamiento serán: idUser, caract1, val1_1, val1_2, val1_3, val1_4, val1_5, caract2, val2_1, val2_2, val2_3, val2_4, val2_5, caract3, val3_1, val3_2, val3_3, val3_4, val3_5, caract4, val4_1, val4_2, val4_3, val4_4, val4_5
Nombre	Campos de las tablas de entrenamiento

RCM-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El método que adquiriera un átomo de información deberá tener 3 parámetros de entrada, con los siguientes tipos Usuario, String y String
Nombre	Parámetros método de adquisición de un átomo de información

RCM-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El método que infiera un átomo de información deberá tener 3 parámetros de entrada, con los siguientes tipos Usuario, String y JList
Nombre	Parámetros método de inferencia de un átomo de información

RCM-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El método que infiera varios átomos de información deberá tener 3 parámetros de entrada, con los siguientes tipos Usuario, String y JList
Nombre	Parámetros método de inferencia de varios átomos de información

RCM-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El método que comprueba un átomo de información deberá tener 4 parámetros de entrada, con los siguientes tipos Usuario, String, String y JList
Nombre	Parámetros método de comprobación de un átomo de información

RCM-007	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El método que cierre la sesión tener 2 parámetros de entrada, con los siguientes tipos Usuario y JList
Nombre	Parámetros método de cierre de sesión

RDC-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El lenguaje empelado para la documentación del sistema será el castellano
Nombre	Lenguaje castellano

RDC-002	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se elaborará un manual de usuario de la aplicación
Nombre	Manual de usuario

RDC-003	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Los comentarios deberán describir todas las funciones del código fuente de manera clara y concisa
Nombre	Comentarios del código fuente

RDC-004	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El tipo letra empleado para toda la documentación del proyecto es Times New Roman
Nombre	Tipo de letra en la documentación

RDC-005	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Desarrolladores
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El tamaño letra empleado para toda la documentación del proyecto es de 11 puntos
Nombre	Tamaño de letra en la documentación

RDC-006	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	En la redacción del manual de usuario se emplearán ejemplos de uso, evitando la utilización de tecnicismos que no sean de dominio público. En el caso de que fuera imprescindible su uso habría que explicar su significado en el mismo manual.
Nombre	Legibilidad del manual de usuario

RSG-001	
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Fuente	Cliente
Estabilidad	Estable
Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Para acceder a todas las funcionalidades de la aplicación será necesario proporcionar datos tales como la máquina, el usuario, la contraseña, etc.
Nombre	Ejecución de la aplicación

Pruebas de validación

El campo ID hace referencia al identificador unívoco de la prueba. El formato a seguir es el siguiente: PR-<números>, donde el campo <números> son números enteros consecutivos incrementados en una unidad.

PR-001	
Nombre	Obtención de un átomo de información del usuario actual
Descripción	Cargar en la aplicación una característica y un valor para el usuario actual
Elementos relacionados	RCP-005, RRS-001
Especificaciones de entrada	Selección de una característica y un valor. Pulsar el botón “Insertar”
Especificaciones de salida	Se mostrará la característica seguida de un guión y el valor en la lista de la pestaña correspondiente
Entorno de prueba	Conexión con la base de datos elegida, y características cargadas en la lista desplegable de selección

PR-002	
Nombre	Guardar la información obtenida del usuario actual
Descripción	Almacenar en la base de datos elegida la información del usuario actual obtenida previamente
Elementos relacionados	RCP-005, RRS-001
Especificaciones de entrada	Pulsar el botón “Guardar”
Especificaciones de salida	Se añadirá un elemento a la lista correspondiente con el siguiente literal “INFORMACIÓN ALMACENADA”
Entorno de prueba	Conexión con la base de datos elegida El usuario actual ha tenido que haber obtenido información de características y valores, quedando patente en la lista de la pestaña de Adquisición

PR-003	
Nombre	Inferir un átomo de información del usuario actual
Descripción	Predecir el valor de la característica seleccionada del usuario actual a partir modelo de usuarios elegido
Elementos relacionados	RCP-006, RRS-001, RRS-003
Especificaciones de entrada	Selección de una característica de la lista. Pulsar el botón “Inferir uno”
Especificaciones de salida	Se añadirá un elemento a la lista correspondiente con el valor y la certeza obtenidos
Entorno de prueba	Conexión con la base de datos elegida e introducidos los datos de configuración del modelo elegido

PR-004	
Nombre	Inferir varios átomos de información del usuario actual
Descripción	Predecir los valores de la característica seleccionada del usuario actual a partir modelo de usuarios elegido
Elementos relacionados	RCP-006, RRS-001, RRS-003
Especificaciones de entrada	Selección de una característica de la lista. Pulsar el botón “Inferir varios”
Especificaciones de salida	Se añadirán elementos a la lista correspondiente con los valores y las certezas de la característica obtenidos
Entorno de prueba	Conexión con la base de datos elegida e introducidos los datos de configuración del modelo elegido

PR-005	
Nombre	Comprobar un átomo de información del usuario actual
Descripción	Comprobación de certeza del valor de la característica del usuario actual con el modelo de usuarios elegido
Elementos relacionados	RCP-006, RRS-001, RRS-003, RRS-004
Especificaciones de entrada	Selección de una característica de la lista y un valor para la misma. Pulsar el botón “Certeza”
Especificaciones de salida	Se añadirá un elemento a la lista correspondiente la certeza de la característica y el valor elegidos
Entorno de prueba	Conexión con la base de datos elegida e introducidos los datos de configuración del modelo elegido

PR-006	
Nombre	Almacenar datos inferidos
Descripción	Inserción de los valores inferidos seleccionados en la base de datos del modelo
Elementos relacionados	RCP-007, RRS-001, RRS-007
Especificaciones de entrada	Pulsar el botón “Retroalimentación positiva” o “Retroalimentación negativa”
Especificaciones de salida	La inferencia seleccionada será almacenada en la base de datos del modelo
Entorno de prueba	<p>Conexión a la base de datos elegida e introducidos los datos de configuración del modelo elegido</p> <p>La lista de inferencias de la pestaña de Consulta no debe estar vacía y debe tener un elemento seleccionado para ser insertado</p>

PR-007	
Nombre	Aprender usuario y cerrar sesión
Descripción	Almacenar la información del usuario actual en la base de datos del modelo y asignar un nuevo usuario a la sesión de la aplicación
Elementos relacionados	RCP-009, RRS-001
Especificaciones de entrada	Pulsar “Cerrar sesión” dentro del menú
Especificaciones de salida	Almacenada la información del usuario actual y asignado nuevo identificador de usuario
Entorno de prueba	Conexión a la base de datos elegida e introducidos los datos de configuración del modelo elegido

PR-008	
Nombre	Entrenamiento del modelo con todos los registros
Descripción	Inserción de todos los registros de la tabla seleccionada en la base de datos del modelo
Elementos relacionados	RCP-010, RRS-001
Especificaciones de entrada	Introducir los datos de configuración, seleccionar el botón “Todos” y pulsar el botón “Iniciar” en la pestaña Entrenamiento
Especificaciones de salida	Presencia de todos los registros de la tabla elegida en el modelo seleccionado
Entorno de prueba	Conexión a la base de datos elegida

PR-009	
Nombre	Entrenamiento del modelo con los registros seleccionados
Descripción	Inserción de todos los registros de la tabla elegida, correspondientes a los identificadores de usuario contenidos en el rango elegido, en el modelo seleccionado
Elementos relacionados	RCP-010, RRS-001, RRS-005
Especificaciones de entrada	Introducir los datos de configuración, seleccionar el botón “Del” y pulsar el botón “Iniciar” en la pestaña Entrenamiento
Especificaciones de salida	Presencia de los registros seleccionados de la tabla elegida en el modelo seleccionado
Entorno de prueba	Conexión a la base de datos elegida

PR-010	
Nombre	Llamadas a los métodos del modelo
Descripción	Comprobar que ejecuta correctamente todos los métodos implementados del modelo
Elementos relacionados	RRS-002
Especificaciones de entrada	Nombres de los métodos del modelo
Especificaciones de salida	Salidas de los métodos del modelo
Entorno de prueba	Conexión a la base de datos elegida, modelo seleccionado y datos de configuración del mismo introducidos

PR-011	
Nombre	Evaluación de usuarios dentro del corpus
Descripción	Selección de un identificador de usuario existente en el modelo; realizar la inferencia de todas las características del mismo, insertando una característica entre cada una de las inferencias
Elementos relacionados	RCP-012, RRS-001, RRS-006
Especificaciones de entrada	Selección del botón “Dentro del corpus” de la pestaña Evaluación
Especificaciones de salida	Datos de la inferencia en el área de texto de la pestaña Evaluación, habiendo aprendido desde 0 hasta n características
Entorno de prueba	Conexión con la base de datos elegida

PR-012	
Nombre	Evaluación de usuarios fuera del corpus
Descripción	Selección de un identificador de usuario no existente en el modelo; realizar la inferencia de todas las características del mismo, insertando una característica entre cada una de las inferencias
Elementos relacionados	RCP-013, RRS-001, RRS-006
Especificaciones de entrada	Selección del botón “Fuera del corpus” de la pestaña Evaluación
Especificaciones de salida	Datos de la inferencia en el área de texto de la pestaña Evaluación, habiendo aprendido desde 0 hasta n características
Entorno de prueba	Conexión con la base de datos elegida

PR-013	
Nombre	Evaluar varios usuarios
Descripción	Selección de dos identificadores de usuario; para cada uno de ellos: realizar la inferencia de todas las características, insertando una característica entre cada una de las inferencias
Elementos relacionados	RCP-012, RCP-013, RRS-006
Especificaciones de entrada	Ninguna
Especificaciones de salida	Datos de la inferencia en el área de texto de la pestaña Evaluación, habiendo aprendido desde 0 hasta n características, para los dos usuarios seleccionados
Entorno de prueba	Conexión con la base de datos elegida

PR-014	
Nombre	Carga de la configuración
Descripción	Obtención de la información contenida en el fichero de configuración y cargada en la aplicación
Elementos relacionados	RCP-003, RRS-002
Especificaciones de entrada	Fichero config.txt en la ruta de ejecución de la aplicación. Dentro del menú seleccionar “Cargar fichero configuración”
Especificaciones de salida	Configuración visible en la pestaña “Definición del modelo”
Entorno de prueba	Ninguno

PR-015	
Nombre	Guardar la configuración
Descripción	Almacenar la información contenida en la pestaña “Definición del modelo” en el fichero config.txt
Elementos relacionados	RCP-003, RRS-002
Especificaciones de entrada	Fichero config.txt en la ruta de ejecución de la aplicación. Pulsar el botón “Guardar en fichero” en la pestaña “Definición del modelo”
Especificaciones de salida	Fichero config.txt en la ruta de ejecución de la aplicación con la información contenida en la pestaña
Entorno de prueba	Ninguno

Casos de uso

A continuación se muestran las descripciones textuales de los casos del uso de la aplicación Entorno de trabajo con modelos de usuario:

CU-001	
Nombre	Introducir nuevos datos
Descripción	Introducir pares de característica-valor asociados al usuario actual en el sistema
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación
Flujo	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación 2. Acceder a la pestaña “Adquisición” 3. Repetir <ol style="list-style-type: none"> i. Seleccionar característica ii. Seleccionar valor iii. Pulsar botón “Insertar”
Postcondiciones	Lista de características del usuario actual no vacía, conteniendo los pares introducidos

CU-002	
Nombre	Eliminar datos
Descripción	Eliminar pares de característica-valor asociados al usuario actual en el sistema
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación con lista de características no vacía
Flujo	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación 2. Acceder a la pestaña “Adquisición” 3. Repetir <ol style="list-style-type: none"> i. Seleccionar par de la lista de características ii. Pulsar botón “Eliminar”
Postcondiciones	Par de característica-valor eliminado de la lista de características del usuario actual

CU-003	
Nombre	Adquirir nuevos datos
Descripción	Insertar en la base de datos en el modelo la información del usuario actual obtenida
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación con lista de características no vacía
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder a la pestaña “Adquisición”3. Pulsar el botón “Guardar”
Postcondiciones	Nuevos datos del usuario actual en el modelo de usuarios

CU-004	
Nombre	Inferir un átomo de información
Descripción	Obtener un valor y una certeza asociados a una característica para el usuario actual
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación y seleccionada una característica en la pestaña “Consulta”
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder a la pestaña “Consulta”3. Seleccionar una característica4. Pulsar el botón “Inferir uno”
Postcondiciones	Mostrado a través de la interfaz el valor y la certeza inferidos

CU-005	
Nombre	Inferir varios átomos de información
Descripción	Obtener una lista de valores y sus respectivas certezas asociados a una característica para el usuario actual
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación y seleccionada una característica en la pestaña “Consulta”
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder a la pestaña “Consulta”3. Seleccionar una característica4. Pulsar el botón “Inferir todos”
Postcondiciones	Mostrado a través de la interfaz el conjunto de valores y sus certezas inferidos

CU-006	
Nombre	Comprobar un átomo de información
Descripción	Obtener una certeza asociada a un valor de una característica para el usuario actual
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación y seleccionadas una característica y un valor en la pestaña “Consulta”
Flujo	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación 2. Acceder a la pestaña “Consulta” 3. Seleccionar una característica 4. Seleccionar un valor para la característica seleccionada 5. Pulsar el botón “Certeza”
Postcondiciones	Mostrado a través de la interfaz la certeza inferida

CU-007	
Nombre	Aprender inferencia
Descripción	Almacenar en la base de datos del modelo de usuarios los datos inferidos seleccionados
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación habiendo inferido información sobre el modelo y con, al menos, un resultado de inferencia seleccionado
Flujo	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación 2. Acceder a la pestaña “Consulta” 3. Seleccionar un resultado de inferencias 4. Pulsar el botón “Retroalimentación positiva” o “Retroalimentación negativa”
Postcondiciones	Información inferida seleccionada almacenada en el modelo de usuarios

CU-008	
Nombre	Cargar fichero de configuración
Descripción	Obtener los datos de configuración del archivo y configurar el sistema con ellos
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación y fichero config.txt con información en la ruta de ejecución de dicha aplicación
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder al “Menú”3. Pinchar en “Carga fichero de configuración”
Postcondiciones	El modelo de la aplicación configurado con la información contenida en el fichero

CU-009	
Nombre	Almacenar datos de configuración
Descripción	Salvar en un fichero los datos de configuración del modelo con el que está trabajando el sistema
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación y datos de configuración del modelo introducidos en el sistema
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder a la pestaña “Definición del modelo”3. Pulsar el botón “Guardar en fichero”
Postcondiciones	Fichero config.txt con la información de configuración en la ruta de ejecución de la aplicación

CU-010	
Nombre	Introducir datos de configuración
Descripción	Rellenar los campos de definición del modelo del sistema
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder a la pestaña “Definición del modelo”3. Seleccionar un tipo de modelo (Java / Oracle)4. Rellenar todos los campos activos5. Pulsar el botón “Aplicar”
Postcondiciones	El modelo de la aplicación configurado con la información contenida en el fichero

CU-011	
Nombre	Entrenar el modelo
Descripción	Almacenar en la base de datos del modelo de usuarios los usuarios seleccionados de la tabla origen
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación. Datos de entrenamiento completos y rango de usuarios seleccionado
Flujo	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación 2. Acceder a la pestaña “Entrenamiento” 3. Rellenar los campos con la información de la base de datos origen 4. Selección de usuarios (“Todos” / Rango) 5. Pulsar el botón “Iniciar”
Postcondiciones	Información de los usuarios seleccionados almacenada en el modelo de usuarios de la aplicación

CU-012	
Nombre	Evaluar usuarios con aprendizaje
Descripción	Inferir los valores y sus certezas de una característica de cada uno de los usuarios seleccionados, aprendiendo una nueva característica entre una inferencia y otra
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación. Datos de evaluación completos y rango de evaluación seleccionado
Flujo	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación 2. Acceder a la pestaña “Evaluación” 3. Si los campos de la información de la base de datos del modelo están vacíos: rellenar 4. Selección del rango de evaluación (“Dentro del corpus” / “Fuera del corpus”) 5. Seleccionar el número de usuarios a evaluar 6. Pulsar el botón “Evaluar”
Postcondiciones	Mostrado a través de la interfaz el identificador del usuario, el número de características aprendidas en cada caso y, si ha tenido éxito, el valor y su certeza o, en caso contrario, sólo el valor

CU-013	
Nombre	Cerrar sesión
Descripción	Almacenar en la base de datos del modelo la información del usuario actual no guardada con anterioridad en la sesión y crear un nuevo usuario
Actores	Usuario
Precondiciones	Usuario actual dado de alta en la aplicación
Flujo	<ol style="list-style-type: none">1. Iniciar sesión en la aplicación2. Acceder al “Menú”3. Pinchar en “Cerrar sesión”
Postcondiciones	Nuevo usuario actual en el sistema

ANEXO C. Manuales de usuario

Manuales de Usuario

Desarrollo de
infraestructuras para el
modelado de usuarios

*Ingeniería
Informática*



Universidad
Carlos III de Madrid


Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

TABLA DE CONTENIDO

1	WEB CRAWLER	4
2	ENTORNO DE TRABAJO PARA MODELOS DE USUARIO.....	5
2.1	Obtención de información.....	6
2.2	Almacenamiento de información	9
2.3	Introducir nuevos datos	9
2.4	Eliminar datos	11
2.5	Aprendizaje de nuevos datos.....	12
2.6	Inferir un átomo de información	12
2.7	Inferir varios átomos de información.....	13
2.8	Comprobar un átomo de información	14
2.9	Aprender información inferida.....	14
2.10	Entrenamiento	16
2.11	Evaluación.....	17
2.12	Cerrar sesión.....	18

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009


1 WEB CRAWLER

El usuario que desee ejecutar la aplicación Web Crawler debe abrir el entorno de desarrollo Eclipse y ejecutar el proyecto correspondiente al Web Crawler.

Para su correcto funcionamiento es necesario conocer los posibles parámetros que acepta el programa

- **URL:** dirección Web, dentro del contexto de:
<http://es-la.facebook.com/directory/people/>, que servirá como página de partida de la ejecución
- **Cualquier otro parámetro:** si se introduce como parámetro cualquier otra cadena de caracteres o números, la aplicación se ejecutará con normalidad, teniendo como páginas de partida las almacenadas en la base de datos, más concretamente, en la tabla *newsURLs*.

A medida que va avanzando la ejecución del programa se irán almacenando los perfiles visitados en la tabla correspondiente, además de ir mostrando, por la consola, una traza con todas las direcciones de las Webs que va visitando.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

2 ENTORNO DE TRABAJO PARA MODELOS DE USUARIO

El presente manual recoge la funcionalidad ofrecida a un usuario que ejecute la aplicación:

- **Obtención de información de configuración:** permite adquirir la información de configuración del modelo a partir de un fichero de texto o bien introduciéndola en la interfaz.
- **Almacenamiento de información de configuración:** guarda la información de configuración del modelo de la aplicación en un fichero de texto.
- **Introducir nuevos datos:** ofrece al usuario la posibilidad de cargar en el sistema información de características y sus valores correspondientes al usuario actual.
- **Eliminar datos:** permite al usuario borrar información del usuario actual, introducida previamente.
- **Aprendizaje de nuevos datos:** podrá almacenar en el modelo la nueva información obtenida del usuario actual.
- **Inferir un átomo de información:** permite al usuario obtener un valor y una certeza para una característica dada del usuario actual.
- **Inferir varios átomos de información:** obtiene una lista de valores y sus correspondientes certezas para una característica dada del usuario actual.
- **Comprobar un átomo de información:** permite al usuario obtener una certeza para un valor de una característica dada del usuario actual.
- **Aprender información inferida:** se almacena la información del usuario actual inferida previamente.
- **Entrenamiento:** obtiene información de una base de datos para enriquecer el modelo con nuevos datos, de forma masiva.
- **Evaluación:** para cada usuario seleccionado irá aprendiendo una característica y estimando las certezas para cada valor de una característica seleccionada al azar (distinta a la aprendida), conociendo si el valor va acortando la certeza si encaja o si va encajando sino encajaba.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		

19 de Octubre de 2009

- **Cerrar sesión:** permite al usuario cerrar la sesión en la aplicación y aprender la experiencia adquirida durante la misma, si no lo había hecho previamente; así como iniciar sesión con un nuevo usuario en el sistema.

2.1 Obtención de información

Existen dos formas de obtener la información de configuración del modelo: mediante un fichero de configuración o introduciendo los datos a mano en la interfaz.

2.1.1 Fichero de configuración

Antes de proceder a la obtención de información a través del fichero de configuración hay que asegurarse que dentro de la ruta de ejecución de la aplicación existe un fichero de texto “config.txt” con el siguiente formato (se muestra primero el formato para el tipo de modelo Oracle y después para el tipo de modelo Java):

```
Tipo modelo: Oracle

Host:

Usuario:

Contraseña:

Nombre del paquete:

Nombre de la tabla:


1.Método añadir característica:
2.Adquirir un átomo de información:
3.Inferir un átomo de información:
4.Comprobar un átomo de información:
5.Inferir varios átomos de información:
6.Cerrar sesión (aprendiendo):
```

Ilustración 1. Formato fichero, modelo Oracle

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

```
Tipo modelo: Oracle

1.Método añadir característica:
2.Adquirir un átomo de información:
3.Inferir un átomo de información:
4.Comprobar un átomo de información:
5.Inferir varios átomos de información:
6.Cerrar sesión (aprendiendo):
```

Ilustración 2. Formato fichero, modelo Java

Ahora se puede acceder al Menú y pulsar la opción “Carga fichero configuración”, tras lo cual se introducirán los datos contenidos en el fichero en la pestaña “Definición del modelo” para que éstos sean visibles y modificables.

2.1.2 Interfaz, pestaña “Definición del modelo”

Se ha de acceder a la pestaña “Definición del modelo”, mostrada en la siguiente ilustración:

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

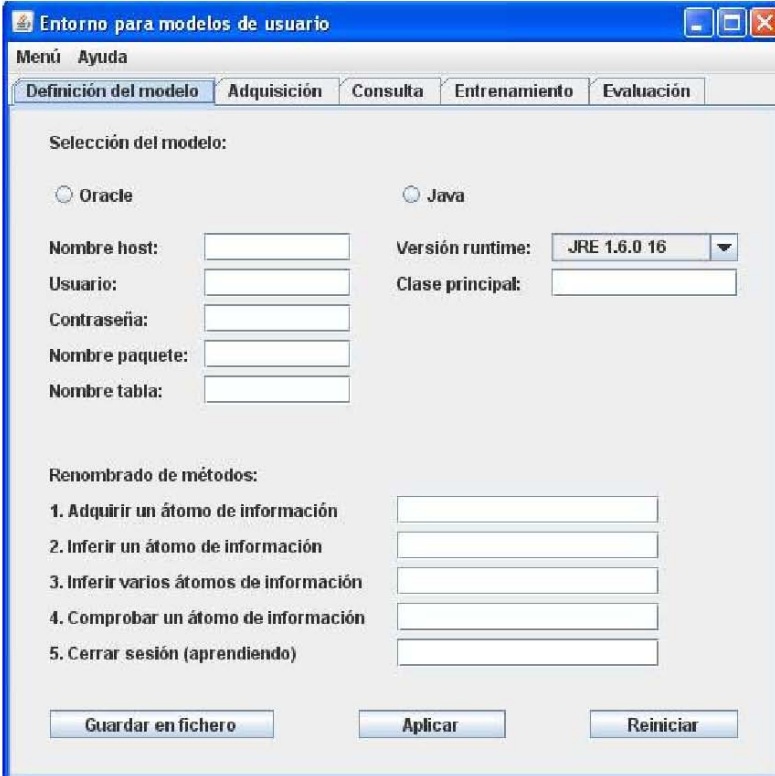


Ilustración 3. Interfaz, pestaña Definición del modelo

Ahora se debe seleccionar el tipo de modelo de usuario (Oracle / Java) y rellenar los campos correspondientes al elegido. Además, se deben introducir los nombres de los métodos implementados en el modelo seleccionado.

Una vez introducidos todos los campos necesarios se pulsará el botón “Aplicar” situado en la zona inferior central de la ventana.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

2.2 Almacenamiento de información

Para proceder al almacenamiento de información de configuración es necesario haber introducido los datos de configuración del modelo en la pestaña “Definición del modelo” (ver Obtención de información)

Una vez que se tiene la pestaña de “Definición del modelo” con datos sobre el mismo se puede pulsar el botón “Guardar en fichero” en la parte inferior izquierda de la misma pestaña y se generará un fichero config.txt en la ruta de ejecución de la aplicación.

2.3 Introducir nuevos datos

Cuando un usuario inicia la aplicación, aparecerá por defecto la pestaña de “Adquisición”, en la cual se va a tener que trabajar a la hora de introducir nueva información acerca del usuario actual (Ilustración 3).

Primeramente se ha de seleccionar una característica de la lista desplegable que se encuentra justo al lado del rótulo con el mismo nombre y, una vez elegida, seleccionar un valor para ésta, de la misma forma, en la lista desplegable que se encuentra justo debajo de la anterior.

En el caso de que se quiera añadir una nueva características, que no se encuentre previamente en la lista proporcionada y, por tanto, tampoco en la base de datos del modelo. Se tendrá que seleccionar la característica con nombre “Nuevo” de tal forma que se abrirá una nueva ventana donde se solicitará el nuevo nombre de la característica y, acto seguido, otra nueva ventana donde poder introducir el nuevo valor para dicha característica. Al finalizar la inserción de los nuevos nombres, éstos quedarán seleccionados en las respectivas listas desplegables.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

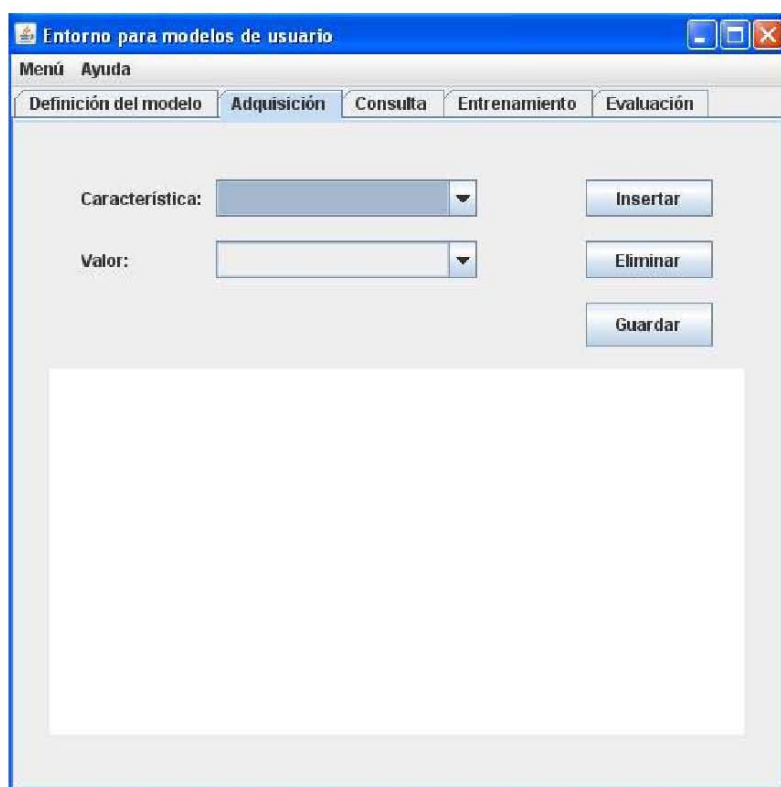



Ilustración 4. Interfaz, pestaña Adquisición

Una vez que están seleccionados la característica y su valor correspondiente bastará con pulsar el botón “Insertar” situado a la derecha de la ventana para que se añada la información seleccionada a la lista de características y valores del usuario actual.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

2.4 Eliminar datos

Para el proceso de eliminación de datos sobre el usuario actual es necesario que se haya insertado previamente nuevos datos sobre dicho usuario, es decir, que la lista de características y valores de dicho usuario no se encuentre vacía. Tal y como se puede observar en la siguiente ilustración, en la lista de la parte inferior de la ventana:

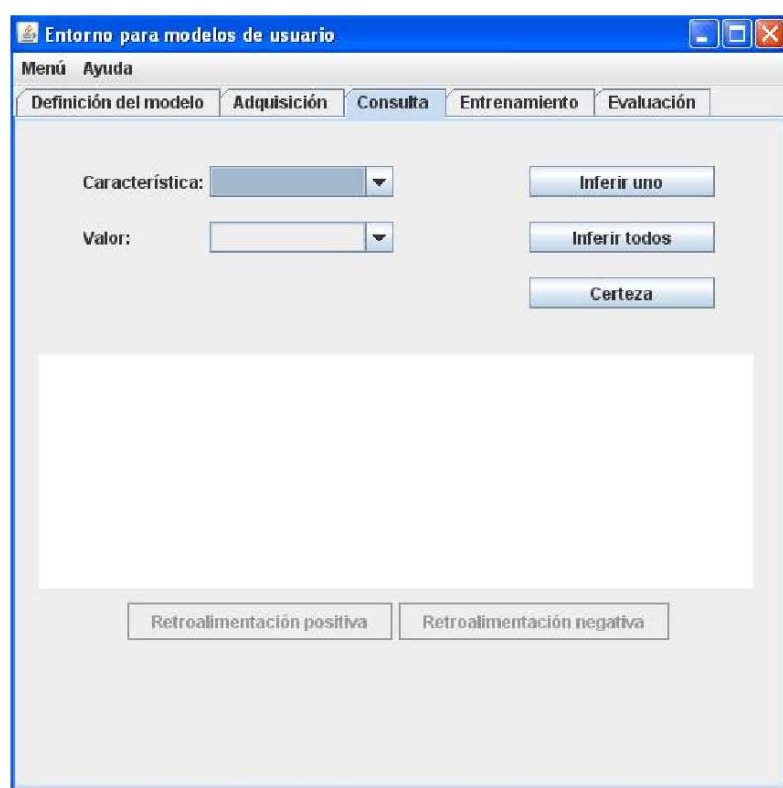


Ilustración 5. Interfaz, pestaña Adquisición

Se seleccionará de la lista de características y valores de la parte inferior de la ventana un par de característica-valor, el que se desee eliminar.

Una vez seleccionado un elemento de la lista se debe pulsar el botón “Eliminar” situado a la derecha de la ventana para que la información seleccionada sea eliminada de la lista de características y valores del usuario actual.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		

2.5 Aprendizaje de nuevos datos

Para el proceso del aprendizaje de nuevos datos sobre el usuario actual es necesario que se haya insertado previamente nuevos datos sobre dicho usuario, es decir, que la lista de características y valores de dicho usuario no se encuentre vacía. Tal y como se puede observar en la Ilustración 5, en la lista de la parte inferior de la ventana.

Además, también será necesario haber introducido los datos de configuración del modelo, así como el nombre del método de “Adquirir un átomo de información”. Esto puede hacerse bien a través de un fichero de configuración, bien a través de la pestaña “Definición del modelo” (ver Obtención de información).


Una vez que comprobado que la lista de características y valores del usuario actual no se encuentra vacía bastará con pulsar el botón “Guardar” situado a la derecha de la ventana para que se añada la información seleccionada a la lista de características y valores del usuario actual.

2.6 Inferir un átomo de información

Se ha de acceder a la pestaña “Consulta”, mostrada en la siguiente ilustración y seleccionar una característica de la lista desplegable que se encuentra justo al lado del rótulo con el mismo nombre.

Además, también será necesario haber introducido los datos de configuración del modelo, así como el nombre del método de “Inferir un átomo de información”. Esto puede hacerse bien a través de un fichero de configuración, bien a través de la pestaña “Definición del modelo” (ver Obtención de información).

Una vez se tiene la característica seleccionada se pulsará el botón “Inferir uno” situado en la zona derecha de la ventana. Tras esto deberá aparecer una nueva línea en la lista de la parte inferior de la ventana, mostrando el valor y la certeza inferidos para la característica seleccionada.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

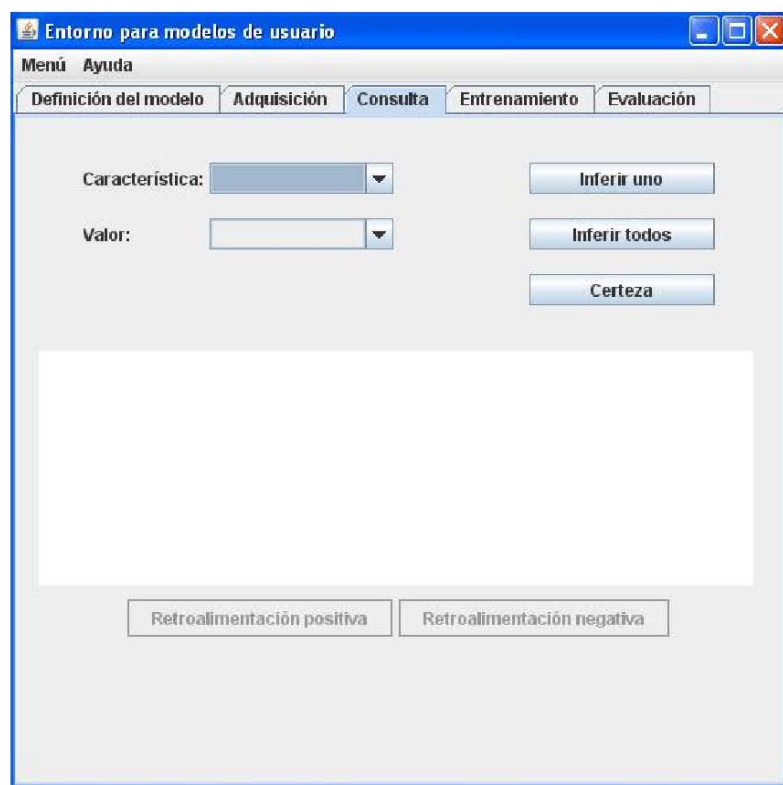


Ilustración 6. Interfaz, pestaña Consulta

2.7 Inferir varios átomos de información

Se ha de acceder a la pestaña “Consulta”, mostrada en la siguiente ilustración y seleccionar una característica de la lista desplegable que se encuentra justo al lado del rótulo con el mismo nombre.

Además, también será necesario haber introducido los datos de configuración del modelo, así como el nombre del método de “Inferir varios átomos de información”. Esto puede hacerse bien a través de un fichero de configuración, bien a través de la pestaña “Definición del modelo” (ver Obtención de información).

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		

Una vez se tiene la característica seleccionada se pulsará el botón “Inferir todos” situado en la zona derecha de la ventana. Tras esto deberán aparecer tantas líneas como valores posea la característica en la lista de la parte inferior de la ventana, mostrando los valores y sus respectivas certezas inferidos para la característica seleccionada.

2.8 Comprobar un átomo de información

Se ha de acceder a la pestaña “Consulta”, mostrada en la siguiente ilustración y seleccionar una característica de la lista desplegable que se encuentra justo al lado del rótulo con el mismo nombre. De la misma manera se seleccionará un valor para la característica seleccionada.

Además, también será necesario haber introducido los datos de configuración del modelo, así como el nombre del método de “Comprobar un átomo de información”. Esto puede hacerse bien a través de un fichero de configuración, bien a través de la pestaña “Definición del modelo” (ver Obtención de información).

Una vez seleccionados la característica y su valor se pulsará el botón “Certeza” situado en la zona derecha de la ventana, tras lo cual deberá aparecer una nueva línea en la lista de la parte inferior de la ventana, mostrando la certeza inferida para el valor de la característica seleccionada.

2.9 Aprender información inferida

Para el proceso del aprendizaje de información inferida es necesario que se haya inferido información previamente sobre dicho usuario, es decir, que la lista de resultados de la pestaña “Adquisición” no se encuentre vacía. Tal y como se puede observar en la Ilustración 7, en la lista de la parte inferior de la ventana.

Además, también será necesario haber introducido los datos de configuración del modelo, así como el nombre del método de “Adquirir un átomo de información”. Esto puede hacerse bien a través de un fichero de configuración, bien a través de la pestaña “Definición del modelo” (ver Obtención de información).

Una vez comprobado que se cumplen con los requisitos anteriores se seleccionará una línea de la lista de resultados mencionada (la que se quiera insertar) y se pulsará el botón “Retroalimentación positiva” en el caso de que se esté de acuerdo con el valor inferido, puesto que se almacenará con una certeza positiva o bien el botón “Retroalimentación negativa” si no se está de acuerdo con el valor inferido, almacenándose con certeza negativa.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

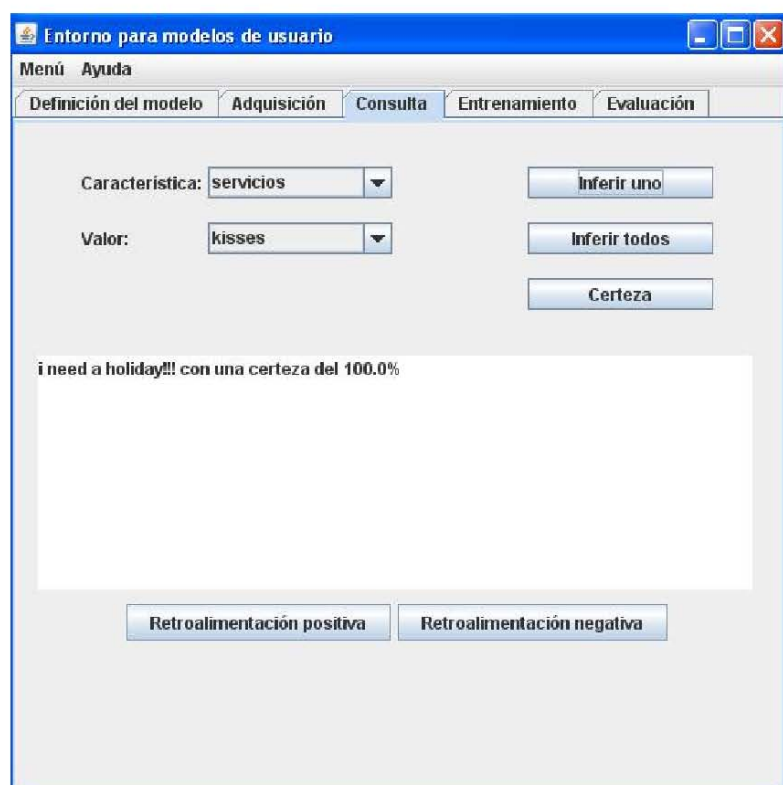


Ilustración 7. Interfaz, pestaña Consulta, con inferencia realizada

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

2.10 Entrenamiento

Se ha de acceder a la pestaña “Entrenamiento”, mostrada en la siguiente ilustración:



Ilustración 8. Interfaz, pestaña Entrenamiento

Ahora se debe introducir la información de la base de datos desde la cual se quieren adquirir los datos, así como seleccionar el rango de usuarios de los cuales se quiere aprender: la tabla completa o bien un rango de identificadores de usuario determinado.

Una vez que se tienen todos los datos introducidos se puede pulsar el botón “Iniciar” en la parte inferior derecha de la misma pestaña y se almacenarán las experiencias de los usuarios seleccionados en la base de datos del modelo.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

2.11 Evaluación

Se ha de acceder a la pestaña “Evaluación”, mostrada en la siguiente ilustración:

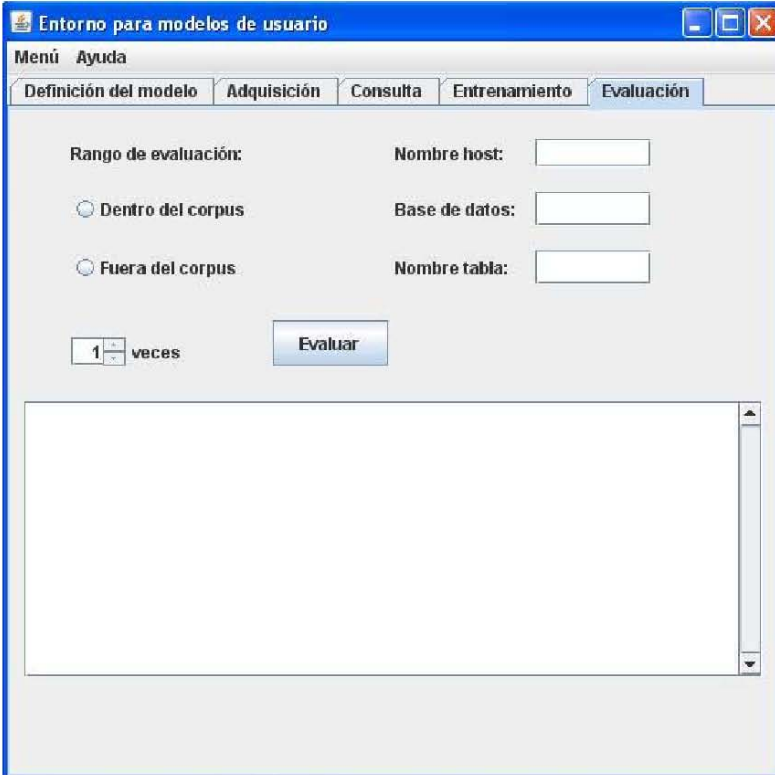


Ilustración 9. Interfaz, pestaña Evaluación

Ahora se debe introducir la información de la base de datos donde se encuentran los datos que se quieren evaluar (en el caso de que no se encuentren ya rellenos), así como seleccionar el rango donde se quiere realizar la evaluación: dentro del corpus (con usuarios ya existentes en el modelo) o fuera del corpus (con usuarios no existentes en el modelo). También se debe elegir el número de usuarios que se va a evaluar en el rango seleccionado.

Una vez que se tienen todos los datos introducidos se puede pulsar el botón “Evaluar” en la parte central de la misma pestaña. Tras esto, se mostrarán los resultados de la evaluación en el área de texto situada en la parte inferior de la ventana.

Desarrollo de infraestructuras para el modelado de usuarios	Manuales de Usuario	 Universidad Carlos III de Madrid
Ana del Coso Santos		19 de Octubre de 2009

2.12 Cerrar sesión

Para cerrar la sesión del usuario actual en la aplicación es necesario haber introducido los datos de configuración del modelo, así como el nombre del método de “Adquirir un átomo de información”, para poder almacenar la información no adquirida previamente (ver Obtención de información).

Una vez hechas las comprobaciones anteriores bastará con seleccionar la opción “Cerrar sesión” en el menú para que la información del usuario actual sea aprendida y para que un nuevo usuario inicie sesión en el sistema.

ANEXO D. Manuales de instalación

Para poder ejecutar el programa desarrollado de manera correcta será necesario instalar y configurar una serie de programas, mediante los que el lector podrá comprobar las funcionalidades detalladas en esta memoria.

En este capítulo se describe con todo detalle los pasos a seguir para obtener una estación de trabajo correctamente configurada. Todos los programas necesarios han sido desarrollados para Windows.

El desarrollo de este proyecto ha sido llevado a cabo sobre un equipo con las siguientes características:

- Pentium III CPU 1.40 GHz.
- 512Mb de RAM.
- 40Gb de Disco Duro
- Sistema Operativo: Windows XP SP 2.

A continuación se presenta una lista con las herramientas necesarias, cuya instalación y configuración se detallará a continuación:

- Base de Datos Oracle 9i.
- Máquina Virtual de Java 1.4.2
- Algún IDE (Entorno Integrado de Desarrollo) para Java, en este caso hemos elegido Eclipse.

Base de datos Oracle 9i

A continuación se explica paso a paso (con capturas de pantalla) cómo instalar Oracle 9i en Windows (cualquier versión).

Los tres CDs que componen el programa de instalación se pueden descargar gratuitamente (siempre que no sea con fines lucrativos) desde la propia Web de Oracle: www.oracle.com.

Introducimos en CD 1 de instalación, y seguimos los pasos mostrados a través de las figuras. Una vez seleccionada la ruta de instalación, seleccionamos el producto que deseamos instalar, en nuestro caso, será Oracle Database 9.2.0.1.0.



Figura 1. Instalación de Oracle. Elección de la ruta de instalación

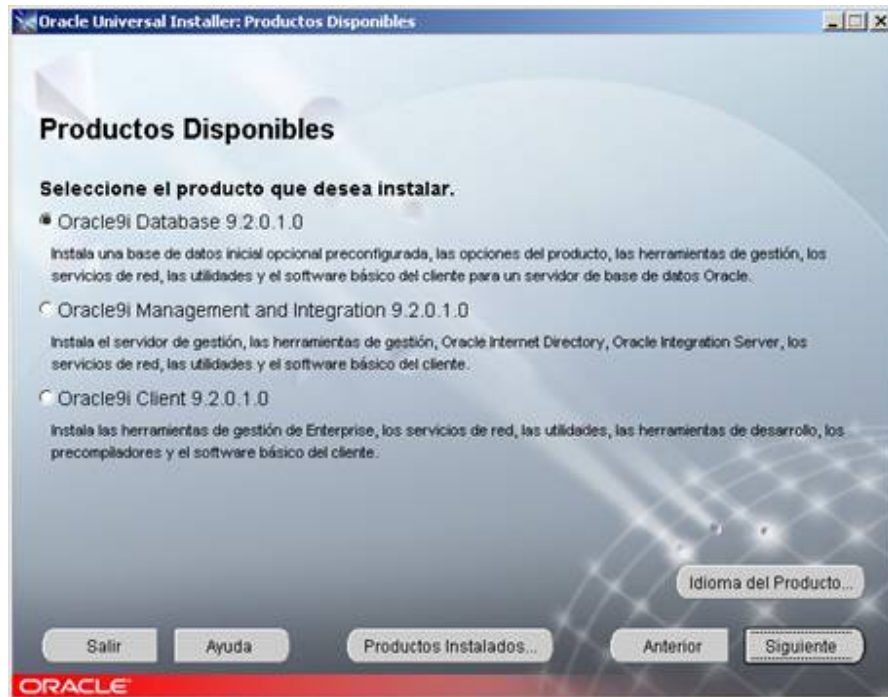


Figura 2. Instalación de Oracle. Selección del producto a instalar.

Después hemos de elegir el tipo de instalación (Enterprise Edition), y en la siguiente ventana que aparezca habrá que especificar la configuración de la base de datos, siendo ésta de Uso general, para que la propia instalación cree una base de datos configurada previamente.

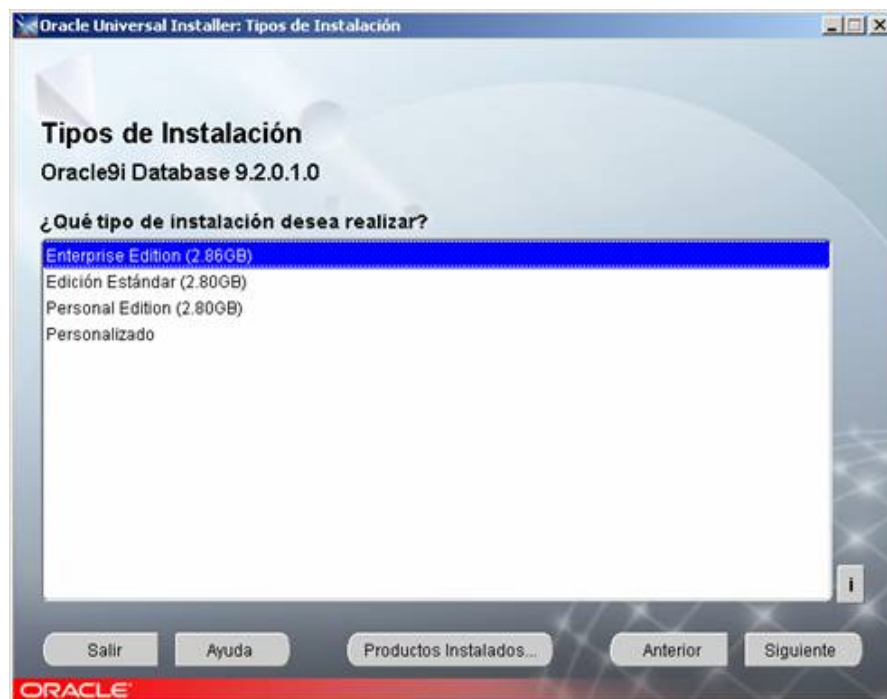


Figura 3. Instalación de Oracle 9i. Elección del tipo de instalación.

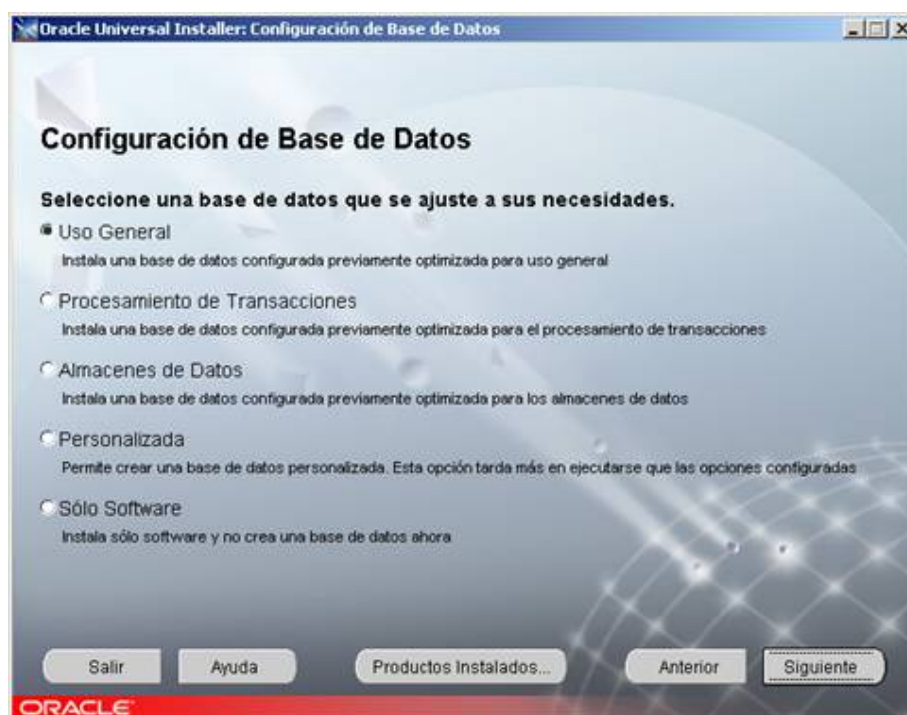


Figura 4. Instalación de Oracle 9i. Configuración de la Base de Datos.

En la siguiente pantalla dejamos el puerto que viene, puesto que esto no afecta a nuestros fines. Acto seguido es necesario introducir el nombre de la base de datos global y el SIDD (identificador interno de Oracle para referenciar la base de datos), ambos con 8 caracteres como máximo. Normalmente coincidirán, pero se pueden modificar.



Figura 5. Instalación de Oracle 9i. Número de puerto



Figura 6. Instalación de Oracle 9i. Elección del identificador de la base de datos.

Ahora se confirma la ruta de instalación, para elegir después el juego de caracteres, usaremos el juego por defecto, que es el asignado por el sistema operativo. Como último paso de la instalación aparece una pantalla donde se especifica el software que se va a instalar, y sólo queda pulsar *'Install'*.



Figura 7. Instalación de Oracle 9i. Confirmación de la ruta de instalación de archivos.



Figura 8. Instalación de Oracle 9i. Elección del juego de caracteres.



Figura 9. Instalación de Oracle 9i. Información sobre el software a instalar.

Una vez completado el proceso de instalación aparece otra pantalla de configuración de la base de datos creada, donde se deben introducir las contraseñas para los usuarios de la base de datos SYS y SYSTEM:

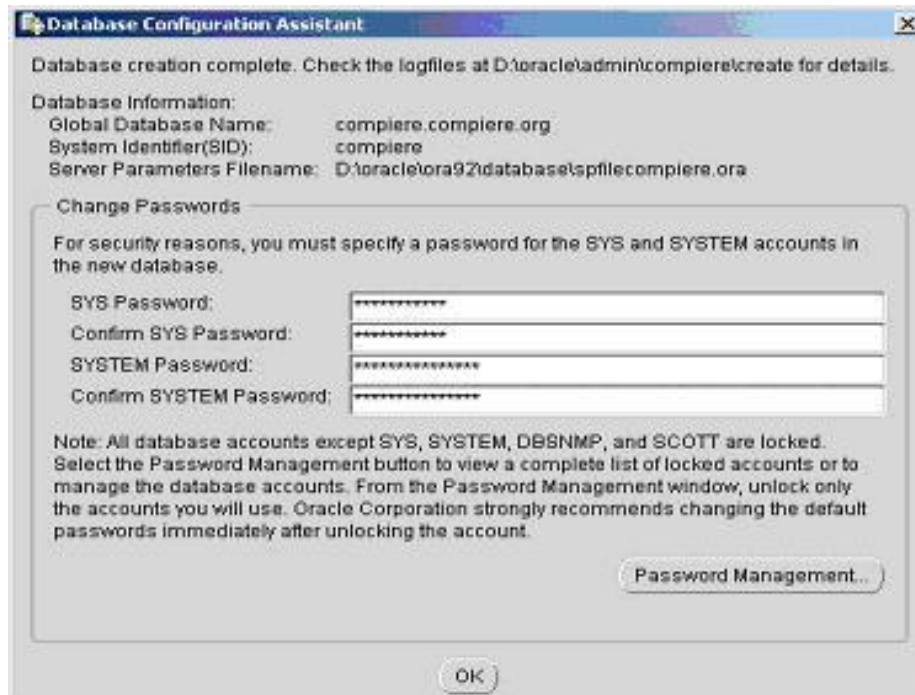


Figura 10. Instalación de Oracle 9i. Inserción de nuevas contraseñas.

Se introducirán los CDs de instalación según vayan siendo requeridos:



Figura 11. Instalación de Oracle 9i. Petición de nuevo CD de instalación.

Máquina Virtual de Java 1.4.2

Para compilar y ejecutar Java se utiliza habitualmente el Java Development Kit (kit de desarrollo de Java) o JDK, desarrollado por Sun Corporation. El JDK tiene dos familias principales: Java 2 Standard Edition o J2SE, orientado a uso personal, y Java 2 Enterprise Edition o J2EE, la edición empresarial.

J2SE se distribuye individualmente o junto con un entorno de edición de código en Java, también de Sun, que se llama NetBeans. A pesar de que NetBeans es un entorno muy potente y amigable, se ha decidido usar Eclipse, cuya instalación se explicará más adelante.

Descargaremos el JDK desde: <http://java.sun.com/j2se/1.4.2/download.html>, pinchando en el enlace que pone “**Download J2SE SDK**”. Una vez descargado será necesario ejecutar el archivo *.exe para comenzar el proceso de instalación.

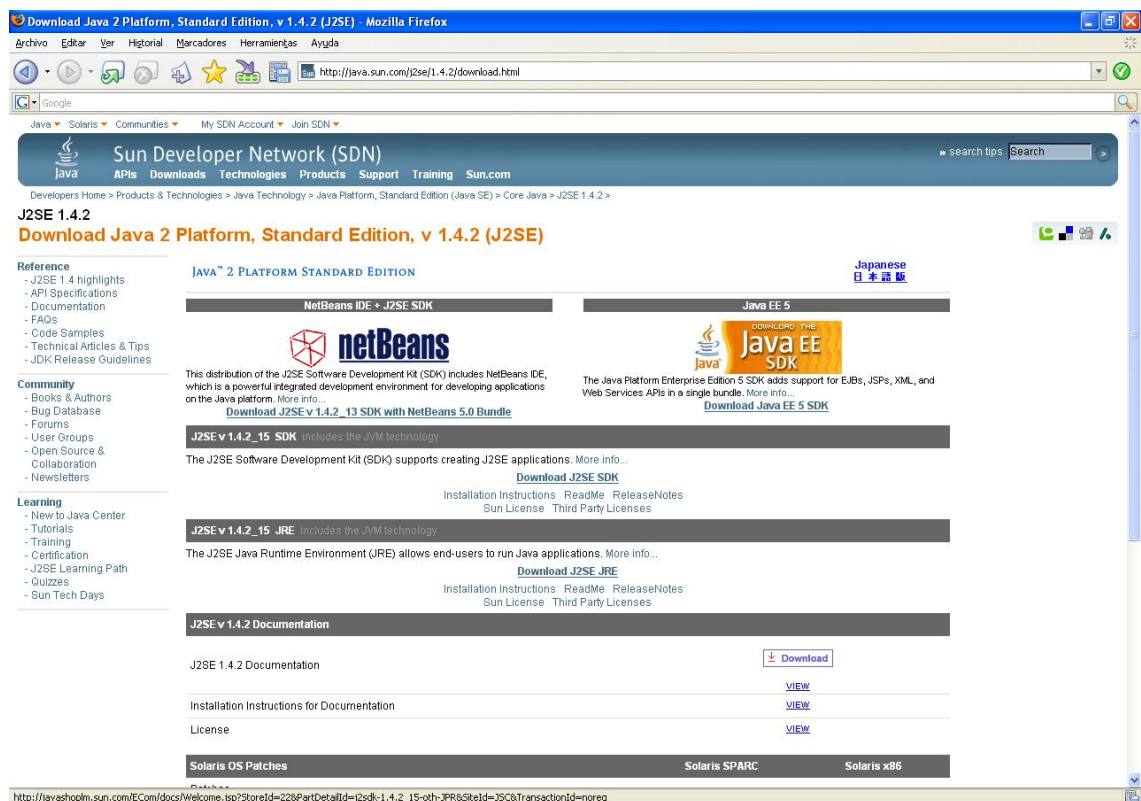


Figura 12. Instalación de Java. Página principal de descargas de Sun.

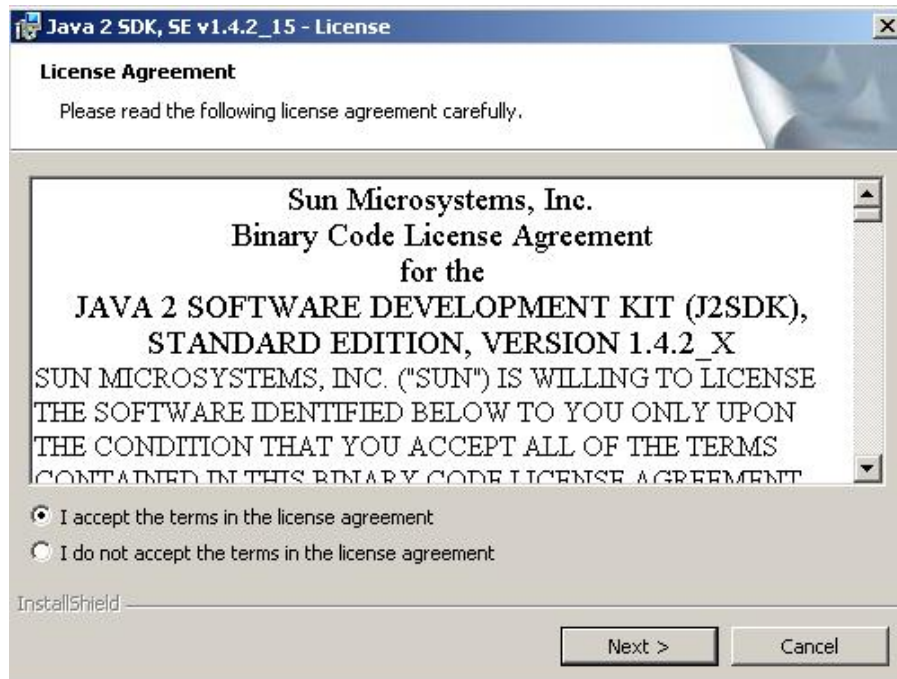


Figura 13. Instalación de Java. Aceptación del contrato de licencia de Sun Microsystems.

Una vez aceptado en contrato de licencia, aparece una pantalla donde se pueden elegir los componentes a instalar, así como el directorio de instalación. Pulsando 'Next' elegimos el/los navegadores en los que queremos registrar Java.

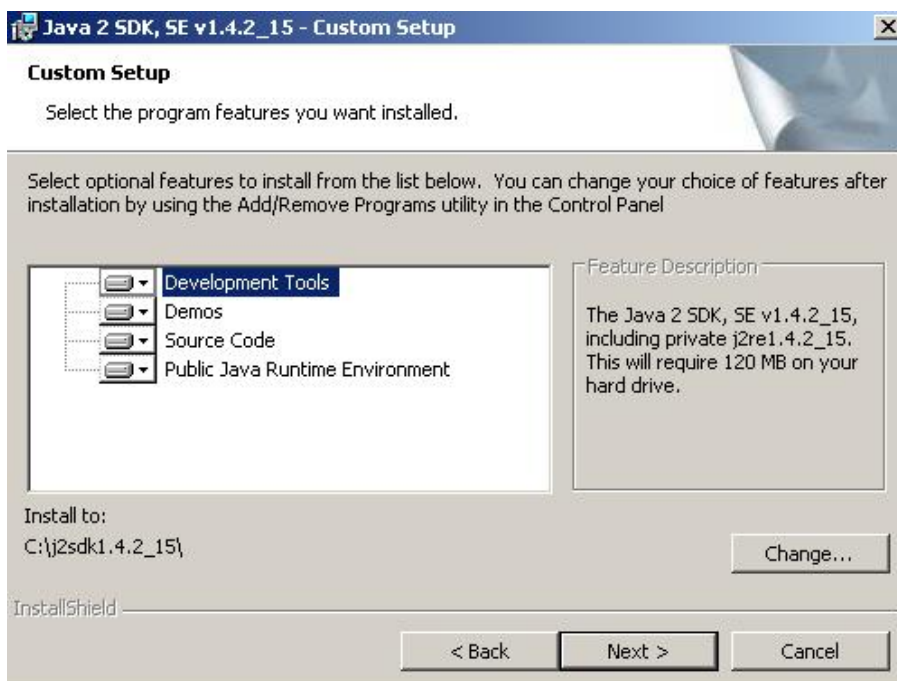


Figura 14. Instalación de Java. Selección del directorio y los componentes.



Figura 15. Instalación de Java. Selección del navegador donde se registrará.

Pulsando el botón '*Install*' el asistente de instalación del programa continuará de manera automática, hasta que pida al usuario reiniciar el sistema.

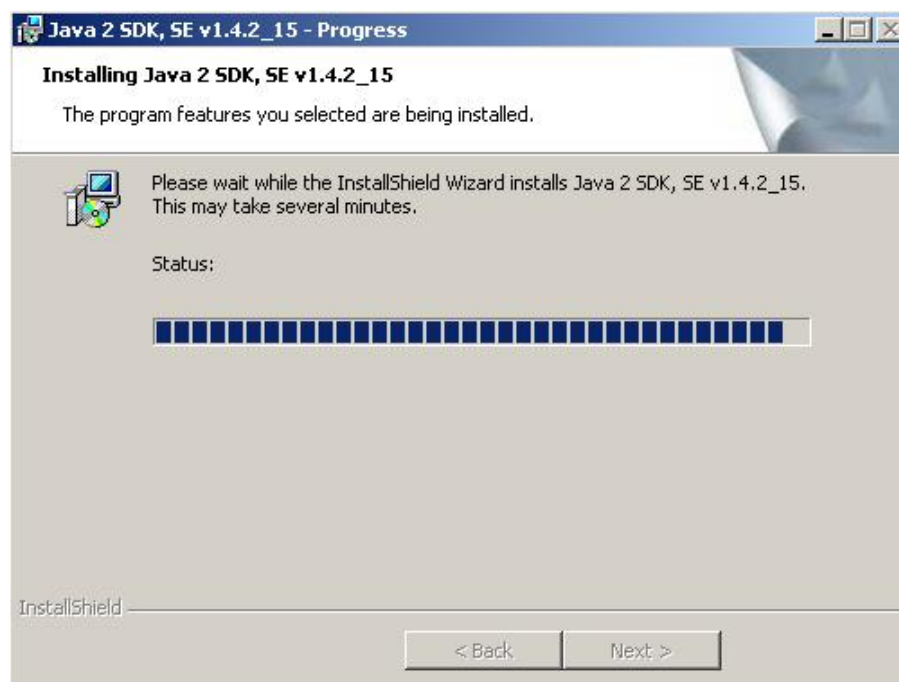


Figura 16. Instalación de Java. Proceso.

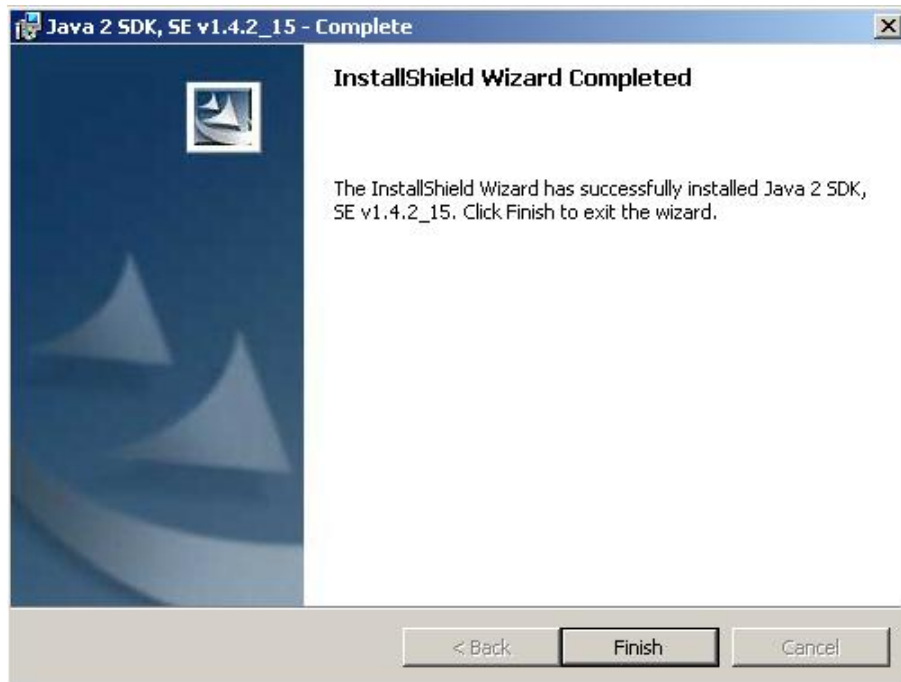


Figura 17. Instalación de Java. Finalizado.



Figura 18. Instalación de Java. Petición de reinicio del sistema.

IDE: Eclipse 3.2

El programa se puede descargar de forma gratuita desde www.eclipse.org un archivo con formato ZIP. Ahora bastará con descomprimir el archivo descargado en la carpeta donde se quiera tener instalado. Para ejecutarlo sólo hay que arrancar el fichero Eclipse.exe.

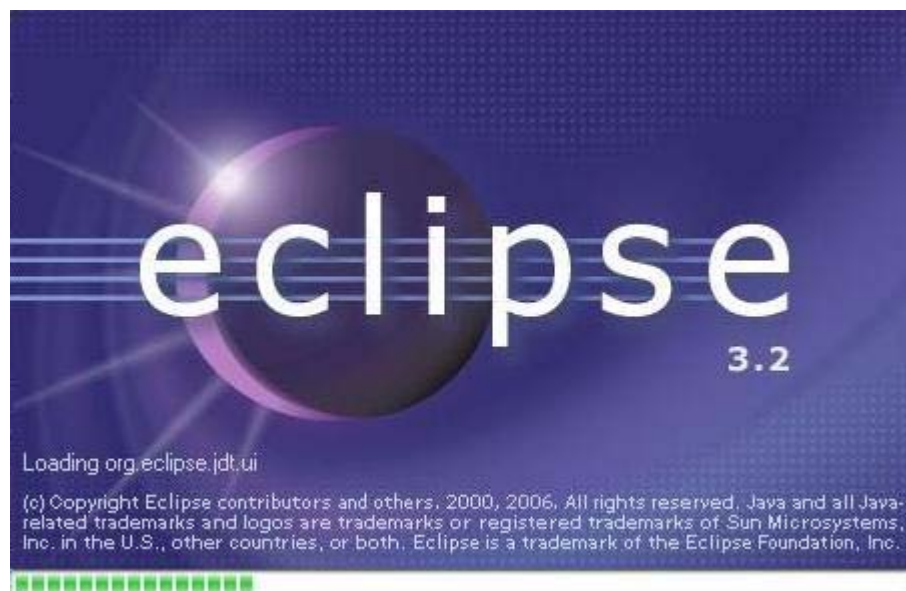


Figura 19. Inicio de Eclipse.

Una vez arrancado lo único que tendremos que introducir es la ruta por defecto donde queremos que el programa guarde los proyectos que creemos.

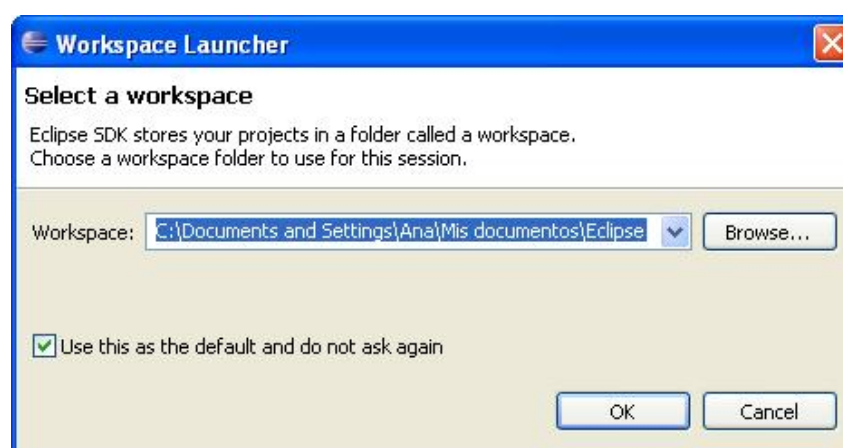


Figura 20. Selección del espacio de trabajo de Eclipse.

Una vez iniciado hay que añadir el driver que van a permitir conectar el sistema con la base de datos creada. Dicho driver se encuentra en el directorio raíz donde se haya instalado Oracle, dentro del directorio jdbc\lib. Ahora bien para insertar el driver en el proyecto de Eclipse, habrá que ir al menú Project → Properties, mostrando la ventana de la Figura 21:

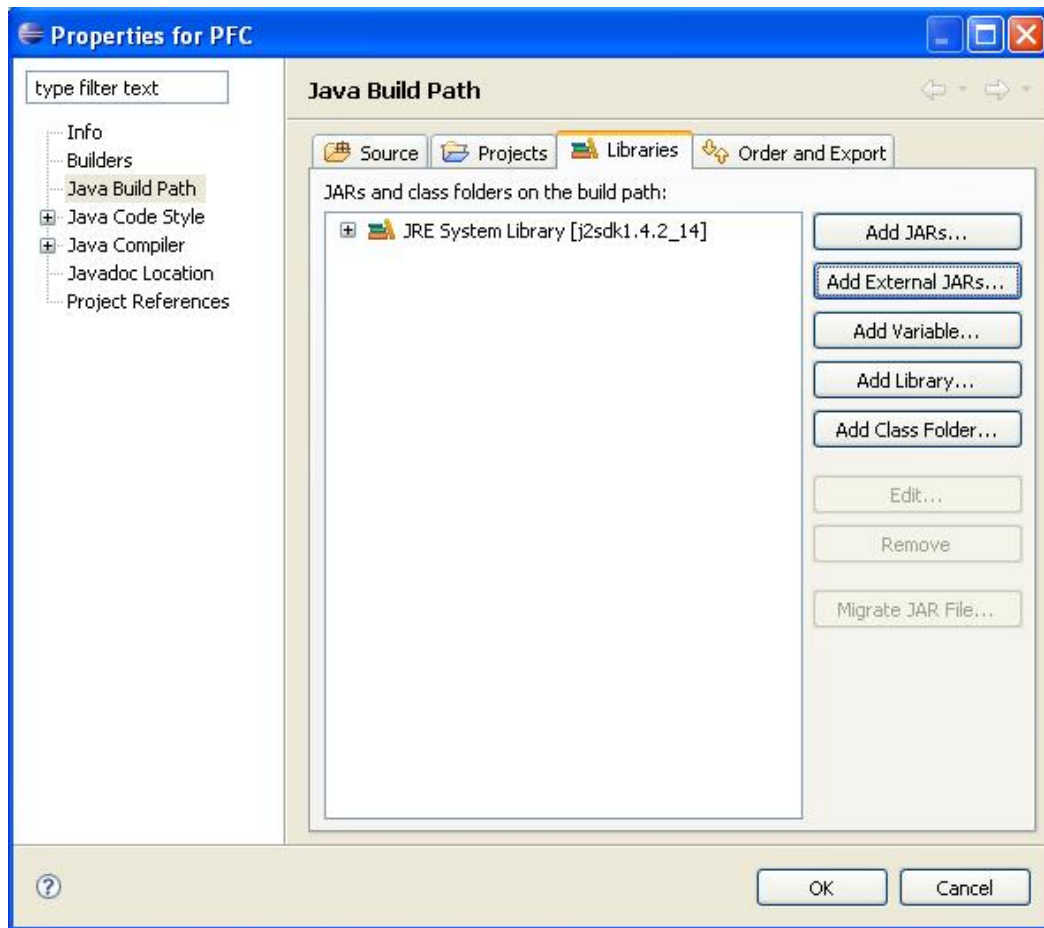


Figura 21. Configuración del proyecto en Eclipse. Añadir el driver para la base de datos.

Aquí deberemos seleccionar el apartado de *Java Build Path* (ruta de ejecución del programa), en la pestaña *Libraries* (librerías) → *Add External JARs*. Se abrirá una nueva ventana desde la que se podrá examinar las carpetas del disco duro; tendremos que llegar hasta la ruta C:\Oracle\jdbc\lib (o lo que es lo mismo la carpeta lib, dentro de la carpeta jdbc del directorio raíz de instalación de Oracle). Ahí seleccionamos el archivo ojdbc14.jar. Una vez

hecho esto ya tendremos asignado el driver correspondiente, quedando la pantalla de la siguiente forma:

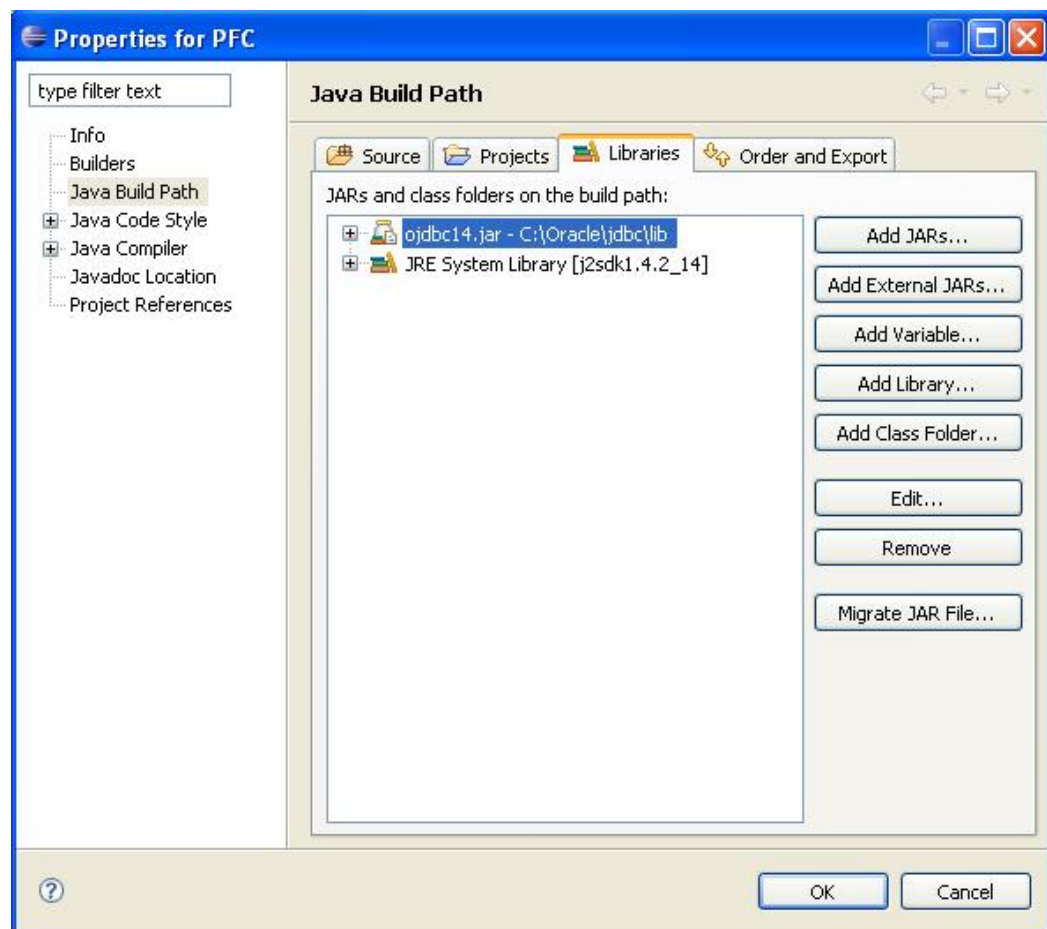


Figura 22. Driver de conexión a la base de datos (ojdbc) asignado.

Ahora ya tendremos configurado Eclipse, por lo que sólo será necesario descomprimir el código del programa en la carpeta seleccionada como entorno de trabajo, y abrir el proyecto PFC en el programa.